MASTER EN SCIENCES INFORMATIQUES

La carte à microprocesseur et son intégration dans une application de paiement électronique

Triniane, Yves

Award date:
1987

Awarding institution:
Universite de Namur

Facultés universitaires Notre-Dame de la Paix NAMUR
Institut d'informatique

La carte à microprocesseur et son intégration dans une application de paiement électronique

Promoteur : Jean Ramaekers

Mémoire présenté par
Yves Triniane

en vue de l'obtention du titre de

LICENCIE ET MAITRE EN INFORMATIQUE

Année académique 1986/1987

Je tiens tout d'abord à remercier Monsieur le Professeur Jean Ramaekers , promoteur de ce mémoire , pour son aide précieuse ainsi que pour sa disponibilité durant cette année .

J'exprime également tous mes remerciements à Monsieur Girardot (Bull CP8) qui m'a organisé efficacement un stage à Trappes , à Monsieur Goire (Bull CP8 , Trappes) pour sa collaboration et les conseils éclairés qu'il m'a prodigués.

De même je remercie toutes les personnes qui m'ont aidé durant ce stage .

Enfin , je témoigne ma gratitude à Madame Stas Cécile pour les judicieux conseils qu'elle m'a apportés dans la réalisation de ce mémoire .

Résumé

Après avoir étudié quelques utilisations de la carte à microprocesseur ; ce mémoire traite , au travers de la description technique de cette carte , des aspects de protection des informations , de la polyvalence et des usages multi-services de la carte .

Vient ensuite , l'étude du scénario de paiement de contact utilisant la carte à microprocesseur . Cette application révèle un accroissement des exigences envers la carte , ce qui oblige celle-ci à évoluer et à satisfaire ces diverses contraintes . La carte à microprocesseur parvient donc à limiter les risques de fraudes et à diminuer le nombre de connexion au central .

Malgré cela , l'application concernée nécessite du système des capacités sécuritaires ou autres bien plus grandes . Les derniers chapitres illustrent les rôles joués par le scénario de l'application , le terminal-caisse et les cartes applicatives, de même que l'importance à attribuer au langage de développement d'application et à son environnement .

Abstract

After studying some smart card utilisations ; this memoir treat , through the technical description of this card , of information protection , polyvalence and multi-applications aspects .

Coming next , the study of payment scenario making use of the smart card . This application reveal increasing demands to the smart card , what bound this one to perform evolutions and to satisfy all the constraints . The smart card is able to restrict the fraud risk and the number of host connections .

Yet , the affected application ask the system to security capacities or much more . The latest chapters illustrate the parts played by the application scrypt , the card reader and the micro secure devices as the importance to attribute to the application development language and to his environment .

PLAN DU MEMOIRE

1. Généralités
- qu'est ce qu'une carte à microprocesseur
- pourquoi une carte à microprocesseur
- typologie des services

2. Les critères d'évaluation de la carte à microprocesseur
- protection de l'information
- polyvalence
- aspects multi-services

3. Description technique
- architecture interne de la puce
- description d'un masque
- les programmes d'application
- cycle de vie de la carte

4. Evaluation de la carte à microprocesseur
- protection de l'information
- polyvalence
- aspect multi-services

5. Le paiement électronique
- la carte et le monde bancaire
- le paiement de contact

6. L'évolution des cartes
- la description du masque M.A.
- les avantages du masque M.A.
- conclusions

7. Le matériel
- le terminal-caisse
- la carte applicative
- la coopération carte applicative - terminal
- conclusions

8. Le langage
- introduction
- les buts et les moyens
- l'interpréteur
- les mémoires de la macromachine
- la macromachine
- le langage
- évaluation

9. Conclusions du mémoire

10. Annexes
- la gestion des échanges carte lecteur-encodeur
- les consignes et les instructions

CHAPITRE 1 :
GENERALITES

1.1. QU'EST CE QU'UNE CARTE A MICROPROCESSEUR ?

1.1.1. La Carte embossée
1.1.2. La Carte à pistes magnétiques
1.1.3. La carte à microcircuit
1.1.3.1. La carte à mémoire simple
1.1.3.2. La carte à logique cablée
1.1.3.3. La carte à microprocesseur

1.2. POURQUOI UNE CARTE A MICROPROCESSEUR ?

1.2.1. Impératifs de sécurité
1.2.2. Impératifs économiques

1.3. TYPOLOGIE DES SERVICES

1.3.1. Le paiement électronique
1.3.2. Le dossier portable
1.3.3. Le contrôle d'accès
1.3.4. La sécurité des systèmes d'information

I. GENERALITES

1.1. QU'EST CE QU'UNE CARTE A MICROPROCESSEUR ?

La carte à microprocesseur fait partie des cartes à microcircuit qui sont elles-mêmes englobées dans la famille des cartes à mémoire. Toute carte à mémoire a pour fonction de mémoriser de l'information et de la protéger de manière à éviter la fraude. Il existe trois types de cartes à mémoire utilisées actuellement :

- la carte embossée,
- la carte à pistes magnétiques,
- la carte à microcircuit.

I.I.I. LA CARTE EMBOSSEE .

Elle est constituée d'une simple carte plastique (PVC) sur laquelle des informations (nom, adresse, numéro de compte du porteur, ...) sont gravées par un procédé d'embossage.

La capacité de mémorisation de cette carte reste très faible .

Le niveau de sécurité offert par cette carte est également très faible. En effet, toute information inscrite sur la carte est directement lisible à l'oeil nu. Il n'est donc pas envisageable d'y graver des informations confidentielles. De plus, il est relativement aisé de fabriquer de fausses cartes.

1.1.2 LA CARTE A PISTES MAGNETIQUES .

Elle est constituée d'une carte plastique (PVC) sur laquelle ont été ajoutées des pistes magnétiques. Les informations peuvent être stockées sur les pistes magnétiques ou sur la carte (par embossage comme précédemment).

La capacité de mémorisation reste relativement faible.

Le niveau de sécurité augmente considérablement par rapport à la carte embossée. En effet, l'accès aux informations mémorisées par les pistes magnétiques ne peut être réalisé qu'en disposant d'un lecteur-encodeur. De

plus les informations peuvent être chiffrées, ce qui garantit un certain degré de confidentialité.

1.1.3 LA CARTE A MICROCIRCUIT .

Elle est constituée d'une carte plastique (PVC) dans laquelle a été inséré un composant électronique (puce). Les informations peuvent être mémorisées dans la puce ou sur le PVC. Les informations stockées dans la puce peuvent être chiffrées ou non.

La capacité de mémorisation de ce type de cartes évolue constamment. Actuellement, elle est de l'ordre de quelques Koctets.

Il existe trois types de cartes à microcircuit :

-la carte à mémoire simple,
-la carte à logique câblée,
-la carte à microprocesseur.

1.1.3.1 LA CARTE A MEMOIRE SIMPLE .

La puce est ici un simple composant mémoire qui ne permet pas de réaliser d'autres fonctions que la lecture ou l'écriture d'informations.

Le niveau de sécurité offert par la carte à mémoire simple est analogue à celui offert par la carte à pistes magnétiques : ici aussi, il suffit de disposer d'un lecteur-encodeur pour avoir accès à l'information stockée dans la puce.

1.1.3.2 LA CARTE A LOGIQUE CABLEE .

La puce est constituée de mémoire et de circuits câblés qui s'interposent entre la mémoire et les lecteurs-encodeurs.

Le niveau de sécurité augmente considérablement puisqu'il n'est plus possible d'accéder directement à la mémoire : il faut, au préalable, demander une autorisation d'accès aux circuits câblés. Ces derniers, possédant une intelligence câblée, sont en mesure d'accorder ou non l'autorisation en fonction du contexte.

1.1.3.3 LA CARTE A MICROPROCESSEUR .

La puce est constituée d'un microprocesseur, d'une mémoire ROM, et d'une mémoire de stockage. Le microprocesseur et la mémoire ROM ont la même fonction que des circuits câblés, c'est-à-dire protéger les accès à la mémoire de stockage. Les cartes à microprocesseur et à logique câblée sont donc semblables au niveau du fonctionnement et au niveau de la sécurité offerte. La différence est que l'intelligence de la carte à microprocesseur n'est pas câblée mais contenue sous forme de programmes dans la mémoire ROM (ce point sera largement détaillé au chapitre 3).

Remarque : pour toutes les cartes à microcircuit, la mémoire de stockage doit être non volatile. On ne peut donc pas utiliser de mémoire RAM traditionnelle. Nous verrons par la suite qu'une mémoire de type PROM convient très bien.

1.2 POURQUOI UNE CARTE A MICROPROCESSEUR ?

Ce sont principalement des impératifs sécuritaires et économiques qui sont à la base de la carte à microprocesseur.

1.2.1 IMPERATIFS DE SECURITE .

Parallèlement à la généralisation de l'utilisation des cartes à mémoire pour le paiement, la fraude sur ces mêmes cartes a connu un taux de croissance particulièrement élevé dans certains pays (surtout aux Etats-Unis). Cette fraude revêt diverses formes :

- vols de cartes, avant ou après leur remise au porteur,
- duplication de cartes à partir d'une carte volée,
- fabrication de fausses cartes,
- .....

1.2.2 IMPERATIFS ECONOMIQUES .

- L'utilisation des modes de paiement traditionnels (chèques, virements, ...) entraîne des charges considérables pour les organismes bancaires et leurs clients car l'enregistrement des paiements implique généralement des traitements manuels. L'utilisation de la carte à microprocesseur permet l'automatisation des traitements et donc une réduction des coûts.

- Il est toujours difficile de garantir une bonne sécurité pour des transactions informatisées s'effectuant loin d'un centre de traitement : il est souvent indispensable d'avoir recours à des réseaux reliant tous les points de service au centre de traitement le plus proche. La carte à microprocesseur, grâce à son intelligence active, prend en charge la gestion de nombreuses procédures de sécurité, ce qui rend souvent facultative l'utilisation de réseaux. Cette solution entraîne de fortes réductions dans le coût des communications dans les pays géographiquement étendus.

I.3. TYPOLOGIE DES SERVICES .

Si la carte à microprocesseur a d'abord été conçue pour résoudre des problèmes de paiement, on s'est très vite aperçu qu'elle constituait aussi un support adéquat pour d'autres types de services. Il existe actuellement quatre domaines privilégiés pour l'utilisation de la carte à microprocesseur :

- le paiement électronique,
- le dossier portable,
- le contrôle d'accès,
- la sécurité des systèmes d'information.

1.3.1 LE PAIEMENT ELECTRONIQUE .

Ce domaine recouvre le télépaiement et le prépaiement.

Le télépaiement consiste à enregistrer auprès de l'organisme de paiement le montant d'une transaction réalisée par le porteur de la carte (retrait d'argent, achat de marchandises, ...). L'enregistrement d'une transaction peut se faire automatiquement via un réseau (système on-line) ou passer par un support informatique (bande magnétique, ...) qui sera transmis périodiquement à l'organisme de paiement (système off-line). Les informations concernant la transaction sont également enregistrées sur la carte pour permettre au porteur de garder une trace des opérations qu'il a effectuées et le cas échéant, de régler d'éventuels différents avec son organisme de paiement.

Le prépaiement sert à supprimer la monnaie pour les paiements de montants faibles. Deux cas peuvent se présenter : le porteur achète des unités de service qui seront consommées au fur et à mesure des besoins (exemple: la carte de téléphone), ou la carte sert d'abonnement électronique c'est-à-dire que la carte est utilisée comme un abonnement classique dont la validité sera contrôlée automatiquement dans chaque point de service (exemple : abonnement autoroute en France).

1.3.2. LE DOSSIER PORTABLE .

Dans ce cas, la carte sert de support pour le stockage d'informations relatives au porteur. Ces informations sont enregistrées soit uniquement sur la carte, soit sur la carte et dans un fichier central. Dans les deux cas, le dossier portable garantit la disponibilité des informations sans avoir recours à des

réseaux. L'apport de la carte dans ce domaine est donc principalement d'ordre économique.

1.3.3 LE CONTROLE D'ACCES.

La carte est utilisée ici comme matérialisation de droit d'accès à des sites (laboratoires, bureaux, ...) ou à des systèmes d'information. Si on considère l'accès à des sites, la carte fait office de clé et le lecteur-encodeur fait office de serrure.

Un premier apport se situe au niveau de la sécurité. Comme nous l'avons déjà signalé, la carte prend en charge la gestion de nombreuses procédures de sécurité qui sont traditionnellement gérées par des médias tels que terminaux, réseaux, ... Dans ce domaine, la carte offre une plus grande résistance à la fraude que la plupart des médias généralement utilisés.

Le second apport se situe au niveau de la convivialité. La carte, grâce à sa capacité de mémorisation, permet de réaliser des connexions automatiques. Par exemple, l'accès à un service via Minitel (France), nécessite un dialogue entre l'utilisateur et le terminal. Pour un service donné, le dialogue est toujours le même. La carte permet de mémoriser les réponses que l'utilisateur doit fournir au terminal et de les communiquer automatiquement au terminal à chaque demande de service.

1.3.4. LA SECURITE DE SYSTEME D'INFORMATION.

Dans le domaine relativement vaste de la sécurité des systèmes d'information, la carte offre des perspectives intéressantes au niveau du contrôle d'accès (cfr supra) et des problèmes de transmission.

Pour assurer une bonne sécurité aux transmissions, il faut d'une part, éviter que des informations soient interceptées et d'autre part, éviter que des informations soient modifiées accidentellement ou par malveillance.

On utilise dans le premier cas la cryptographie (procédé mathématique pour chiffrer un message afin qu'il soit inintelligible pour ceux à qui il n'est pas destiné) et dans le second, la signature électronique (procédé qui permet de garantir qu'un message n'a pas été altéré et qu'il émane d'une source bien définie).

Nous verrons plus tard que la carte permet de réaliser des systèmes de chiffrement et de signature électronique particulièrement fiables.

Remarque : cette classification des types de services n'est pas rigoureuse et n'est donnée qu'à titre indicatif. En pratique, une service peut chevaucher plusieurs des catégories ci-dessus. Par exemple, l'accès à une base de données via un réseau de télécommunications peut mettre en oeuvre simultanément des procédures de sécurité de systèmes d'information et de télépaiement.

CHAPITRE 2 :
LES CRITERES D'EVALUATION DE LA CARTE A MICROPROCESSEUR

2.1. PROTECTION DE L'INFORMATION

- 2.1.1. La sécurité physique
- 2.1.2. La sécurité logique
  - 2.1.2.1. Authentification de l'utilisateur
  - 2.1.2.2. Authentification de la carte
  - 2.1.2.3. Certification
  - 2.1.2.4. Signature électronique
  - 2.1.2.5. Génération de clés de chiffrement

2.2. POLYVALENCE

2.3. ASPECTS MULTI-SERVICES

II. LES CRITERES D'EVALUATION DE LA CARTE A MICROPROCESSEUR

La réalisation de la carte à microprocesseur CP8 a été principalement guidée par trois idées de base :

- la protection de l'information,
- la polyvalence,
- l'aspect multi-services.

2.1. LA PROTECTION DE L'INFORMATION .

La protection de l'information consiste à lutter contre les altérations volontaires ou involontaires de l'information. Il est d'usage de relier le mot sécurité aux actions malveillantes et le mot sûreté aux actions accidentelles.

La sécurité vise à garantir trois qualités de l'information :

- la confidentialite c'est-à-dire le secret,
- l'authenticité c'est-à-dire l'identification des sources (signature),
- l'intégrité c'est-à-dire l'absence de modifications dues à une tentative de fraude.

La sûreté vise à garantir deux qualités de l'information :

-la pérennité c'est-à-dire l'absence de modifications dues à une cause accidentelle (erreur de transmission, ...),
-l'exactitude c'est-à-dire l'absence d'erreurs dans la représentation du réel perçu.

Le domaine d'action de la carte à microprocesseur est principalement la sécurité, mais il ne faut pas oublier que sécurité et sûreté sont souvent intimement liées.

Les mesures de sécurité implémentées grâce à la carte se situent tant au niveau physique qu'au niveau logique.

Remarque : la carte à microprocesseur ne peut garantir à elle seule la sécurité d'une application. Les concepteurs d'application doivent veiller à implémenter certains dispositifs de sécurité complémentaires (gestion de listes noires de cartes volées, ...). Dans la suite de ce mémoire, nous ne parlerons que des procédures de sécurité entièrement (partiellement) gérées par la carte ou par les appareils constituant l'environnement technique de la carte (lecteur-encodeur, ...).

2.1.1. LA SECURITE PHYSIQUE .

Les dispositifs de sécurité implémentés au niveau physique visent à interdire tout accès à la mémoire de stockage non autorisé par le microprocesseur. En d'autres termes, il doit être impossible de lire, écrire ou effacer des informations sans avoir obtenu au préalable l'autorisation du microprocesseur. La sécurité physique est donc directement liée à la technologie de fabrication de la puce et à son architecture interne.

2.1.2 LA SECURITE LOGIQUE .

La sécurité logique recouvre tous les aspects de la sécurité gérés par les programmes de la carte et par les programmes des appareils dialogant avec la carte. Ces programmes implémentent cinq fonctions fondamentales, appelées fonctions de sécurité :

- l'authentification de l'utilisateur,
- l'authentification de la carte,
- la certification,
- la signature électronique,
- la génération de clés de chiffrement.

2.1.2.1. AUTHENTIFICATION DE L'UTILISATEUR

Le probleme posé est de savoir si la carte est utilisée par une personne autorisée (mesure contre les cartes volées). En fonction des opérations à réaliser, cette personne peut être le porteur (possesseur) de la carte ou son émetteur.

La carte authentifie l'utilisateur en comparant une clé proposée par l'utilisateur avec une clé stockée dans la carte.

2.1.2.2. AUTHENTIFICATION DE LA CARTE

Le problème consiste, pour un système externe (ordinateur central ou simple lecteur-encodeur de cartes), à s'assurer qu'il dialogue avec une vraie carte et que cette carte peut accéder au service demandé par l'utilisateur.

Le sytème peut s'en assurer en demandant à la carte d'exécuter un calcul faisant intervenir une clé secrète interne à la carte. Seule une vraie carte peut posséder une clé interne correcte et donc fournir un résultat correct au calcul demandé.

2.1.2.3. CERTIFICATION

La certification consiste à fournir un certificat unique et infraudable relatif à une information déterminée. Ce certificat permet de garantir qu'une information se trouve matériellement à un endroit déterminé de la carte et qu'elle y est correctement inscrite.

Par exemple, à l'occasion d'une transaction monétaire, il faut vérifier que cette opération est bien enregistrée dans la carte, de manière à éviter toute contestation éventuelle à propos de cette transaction.

2.1.2.4. SIGNATURE ELECTRONIQUE

La signature électronique permet de vérifier l'authenticité, l'intégrité et la pérennité des informations au cours d'une transmission.

Le principe est le suivant :

- L'émetteur condense les informations à transmettre en une chaîne de quelques caractères, à laquelle il joint une référence à l'origine du message (signature). Il génère ensuite un certificat en chiffrant la chaîne de caractères et la référence. Ce certificat est alors joint au texte transmis.

- Au départ du texte reçu et de la référence à l'origine du texte, le récepteur calcule un certificat de la même manière qu'à l'émission. Il compare ensuite le certificat qu'il a calculé et celui qu'il a reçu. Si les deux certificats sont identiques, on peut affirmer que le message reçu est authentique et qu'il n'a pas été modifié pendant sa transmission.

2.1.2.5. GENERATION DE CLES DE CHIFFREMENT

Le problème le plus délicat en matière de chiffrement est constitué par le transport des clés. Comment être sûr qu'à aucun moment la clé de chiffrement ne pourra être interceptée? La carte permet de résoudre ce problème en supprimant le transport des clés. L'idée est que lémetteur et le récepteur génèrent chacun les clés de chiffrement, ce qui rend leur transport inutile et leur interception impossible.

La génération de clés comporte deux étapes :

- l'émetteur génère un nombre aléatoire qu'il transmet au récepteur.

- l'émetteur et le récepteur chiffrent le nombre aléatoire en utilisant une clé secrète commune. Sous sa forme chiffrée, le nombre aléatoire constitue une clé de chiffrement.

Grâce à l'utilisation de nombres aléatoires, une seule clé secrète commune peut être utilisée pour générer un nombre quelconque de clés de chiffrement.

Remarque : la génération de clés ne résoud pas complètement le problème du transport de clés dans la mesure où, pour chiffrer le nombre aléatoire, il faut disposer au préalable d'une clé de chiffrement commune. Il faut donc veiller à ce que cette clé ne puisse pas être interceptée pendant son transport. Nous verrons ultérieurement les mesures de sécurité qui protègent les clés communes.

2.2 POLYVALENCE

Comme nous l'avons vu au chapitre 1, la carte peut être utilisée pour divers services (paiement électronique, dossier portable, ...). De plus, au sein d'un même type de service, il peut y avoir des différences considérables. Pour des raisons de coût, il est indispensable que la carte soit assez souple pour pouvoir être utilisée indifféremment pour n'importe quel type de service.

2.3. ASPECT MULTI-SERVICES

Comme la capacité de la mémoire des puces ne cesse de croître, il devient possible d'utiliser une seule carte simultanément pour des services différents. La mémoire disponible est alors partagée pour permettre la mise en oeuvre des différents services.

L'intérêt d'une carte multi-services est évident : elle permet d'éviter aux grands consommateurs de services la possession de plusieurs cartes et, par conséquent, la mémorisation de plusieurs clés secrètes.

CHAPITRE 3 :
DESCRIPTION TECHNIQUE

3.1. ARCHITECTURE INTERNE DE LA PUCE

3.1.1. Le microprocesseur
3.1.2. La mémoire PROM.
3.1.3. La mémoire ROM.
3.1.4. La mémoire RAM.
3.1.5. Interfacage physique de la puce

3.2. DESCRIPTION D'UN MASQUE

3.2.1. Gestion de la mémoire de stockage ROM.
    3.2.1.1. Les zones
    3.2.1.2. Les mots
3.2.2. Fonction logico-mathématique
3.2.3. Gestion des accès à la mémoire PROM.
    3.2.3.1. Lecture et écriture du microprocesseur
    3.2.3.2. Lecture et écriture de l'extérieur
3.2.4. Jeu d'instructions du microprocesseur
    3.2.4.1. Instructions d'initialisation
    3.2.4.2. Instructions basées sur l'algorithme Télépass
    3.2.4.3. Instructions simples
3.2.5. Gestion des échanges entre la carte et le lecteur-encodeur

3.3. LES PROGRAMMES D'APPLICATION

3.3.1. Environnement d'exécution
3.3.2. Principe des communications avec la carte
3.3.3. Exemples typiques de communication avec la carte
    a. Lecture dans une zone libre
    b. Ecriture dans une zone libre
    c. Lecture dans une zone protégée
    d. Ecriture dans une zone protégée

3.4. CYCLE DE VIE DE LA CARTE

3.4.1. Les phases de la vie d'une carte

3.4.1.1. La phase de fabrication

3.4.1.2. La phase d'assemblage

3.4.1.3. La pase de personnalisation

3.4.1.4. La phase active

3.4.1.5. La mort de la carte

3.4.2. Protection de la carte pendant le cycle de vie

III. DESCRIPTION TECHNIQUE

3.1 ARCHITECTURE INTERNE DE LA PUCE .

Bien que nous l'appellions communément microprocesseur, la puce utilisée par la carte est bien plus qu'un simple microprocesseur classique; il s'agit en fait d'un microsystème informatique comportant :

- un microprocesseur 6805 ou 8048,
- une mémoire PROM (Programmable Read Only Memory),
- une mémoire ROM (Read Only Memory),
- une mémoire RAM (Random Acces Memory),
- une interface d'entrée/sortie.

Cet ensemble microprocesseur-mémoires est monolithique, c'est-à-dire intégré dans un seul et même composant. Il est illustré à la figure 3.1.

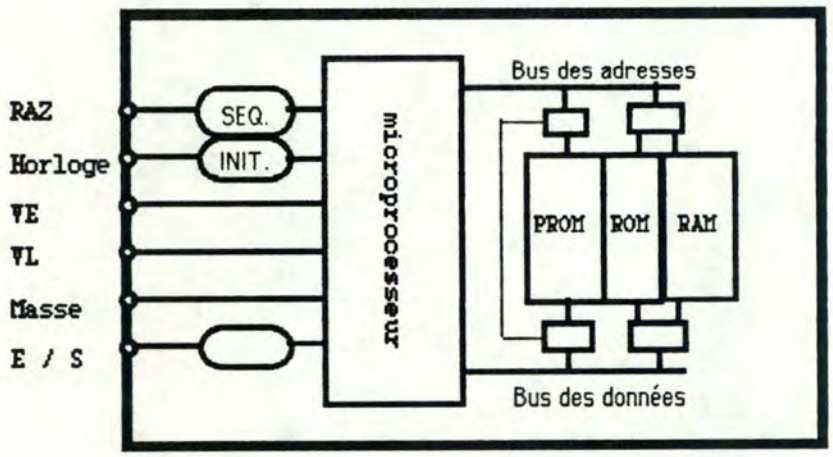


Fig. 3.1 : L'architecture interne de la puce

3.1.1. LE MICROPROCESSEUR .

Ce microprocesseur 8 bits est capable d'exécuter les programmes inscrits dans sa mémoire ROM. Il est autoprogrammable, c'est-à-dire que, contrairement aux architectures classiques, il peut réaliser lui-même l'écriture de sa mémoire PROM à n'importe quel moment.

3.1.2. LA MEMOIRE PROM .

Elle est utilisée pour le stockage des données propres à chaque application (transaction, dossier portable, ...). Dans la suite de ce mémoire, elle sera indifféremment désignée par mémoire PROM ou par mémoire de stockage.

Il faut noter que cette mémoire possède quelques particularités :

- A la fabrication, l'ensemble des bits de cette mémoire est au niveau logique 1. L'écriture consiste à positionner au niveau logique 0 les bits désirés. Tout bit positionner à 0 ne peut plus revenir à l'état 1.

- Pour écrire dans cette mémoire, il faut que la tension d'écriture soit plus élevée que dans le cas d'une lecture des mémoires (d'où la présence dans l'architecture d'un second fil d'alimentation tolérant des tensions plus élevées).

- Cette mémoire PROM, est en fait une mémoire EPROM dont les possibilités d'effacement ne sont pas utilisées.

- A l'avenir, l'EPROM sera remplacée par une EEPROM. Les mémoires EEPROM permettront au processeur d'effacer sa mémoire de façon sélective c'est-à-dire qu'il pourra n'effacer qu'une partie bien définie de la mémoire (par exemple un mot). Il gérera alors l'opération d'effacement comme une opération d'écriture ou de lecture .

3.1.3. LA MEMOIRE ROM

Cette mémoire contient l'intelligence de la carte sous forme de programmes. Cet ensemble de programmes peut varier d'un type de carte à un autre et est appelé MASQUE. Il réalise les fonctionnalités suivantes :

- la gestion de la mémoire de stockage,
- une fonction logico-mathématique,
- la gestion des accès à la mémoire de stockage,
- le jeu d'instructions du microprocesseur,
- la gestion des échanges entre la carte et le monde extérieur.

3.1.4. LA MEMOIRE RAM

C'est la mémoire de travail du microprocesseur. Elle est volatile et ne constitue donc pas un moyen de stockage durable. Elle est utilisée uniquement pour les entrées/sorties et le stockage des informations internes au microprocesseur . Elle est totalement inaccessible de l'extérieur.

3.1.5. INTERFACE PHYSIQUE DE LA PUCE

L'interface physique est constituée de six fils :

- RAZ : une tension appliquée sur ce fil déclenche l'initialisation physique et logique du composant. Suite à cette remise à zéro, la carte envoie des informations sur ses caractéristiques et son état.

- HORLOGE : le microprocesseur n'ayant pas d'horloge interne, celle-ci est externe. Une base de temps de 3,6 MHZ doit être fournie à la carte par ce fil.

- VL : sur ce fil est appliquée la tension d'alimentation du composant, suffisante pour les opérations de lecture des mémoires RAM, ROM et PROM. Cette tension est d'environ 5 volts. La carte ne peut rester sous cette tension plus de 30 secondes.

- VE : sur ce fil est appliquée, à la demande de la carte, la tension nécessaire à la programmation de la PROM. Cette tension est d'environ 21 volts et sa durée maximale de 5 secondes.

- ZERO : fil de masse.

- E/S : c'est par ce fil que transitent les données échangées entre la carte et le monde extérieur. Le mode d'échange est asynchrone - sous forme caractère - et sa vitesse est de 9600 bauds.

Ces six fils sont reliés par soudure à une pastille qui permet d'établir la connexion entre la puce et les systèmes externes (fig. 3.2 et 3.3).

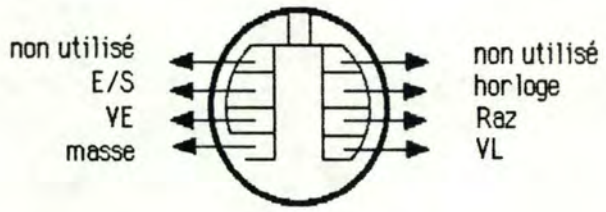


Fig. 3.2 : La pastille

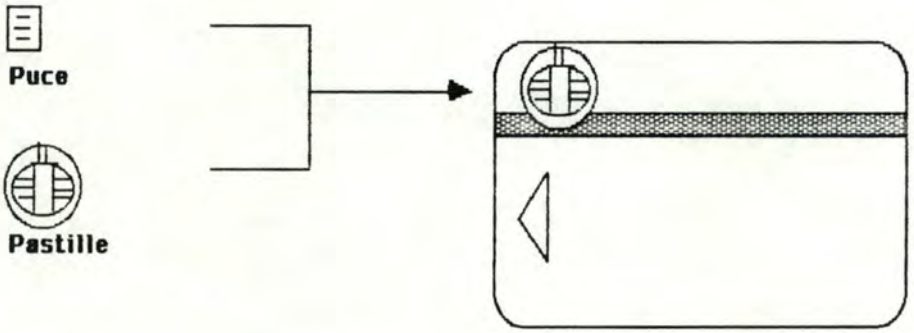


Fig. 3.3 : La carte à microprocesseur

Pour établir la connexion entre la puce et les systèmes externes, la carte doit être insérée dans un connecteur. Celui-ci est spécifiquement conçu pour alimenter électriquement les cartes à microprocesseur et pour permettre le dialogue. Le connecteur est lui-même inséré dans un lecteur-encodeur par lequel transitent toutes les communications. Ces communications seront vues plus en détail au point 3.3. (les programmes d'application).

Remarque : comme nous l'avons déjà signalé, la taille des mémoires ne cesse de croître. Voici un bref historique des types de puce utilisés avec leurs principales caractéristiques.

TYPE DE LA PUCE	MAM01	MAM02	MAM03	MAM04
MICROPROCESSEUR	6805	8048	6805	
MEMOIRE RAM	36 oct.	44 oct.	52 oct.	
MEMOIRE ROM	1600 oct.	2048 oct.	2048 oct.	
MEMOIRE PROM	1024 oct.	1024 oct.	1920 oct.	-> 8192

Fig. 3.4 : Caractéristiques des puces

3.2 DESCRIPTION D'UN MASQUE .

Rappelons que le masque est constitué de l'ensemble des programmes situés en ROM et qu'il réalise essentiellement :

- la gestion de la mémoire de stockage,
- une fonction logico-mathématique,
- la gestion des accès à la mémoire de stockage,
- le jeu d'instructions du microprocesseur,
- la gestion des échanges entre la carte et le monde extérieur.

Nous allons décrire successivement ces cinq aspects du masque. Cette description n'est valable que pour la phase d'utilisation de la carte par son porteur. En effet, nous verrons au paragraphe 3.4 (Cycle de vie de la carte) que la gestion de la mémoire de stockage et la gestion des accès à cette mémoire sont différentes avant et après la mise en service de la carte.

3.2.1. GESTION DE LA MEMOIRE DE STOCKAGE PROM .

La mémoire de stockage est logiquement divisée en 7 zones distinctes qui se subdivisent elles-mêmes en mots.

3.2.1.1 LES ZONES

Le découpage de la mémoire de stockage PROM est donné à la figure 3.5.

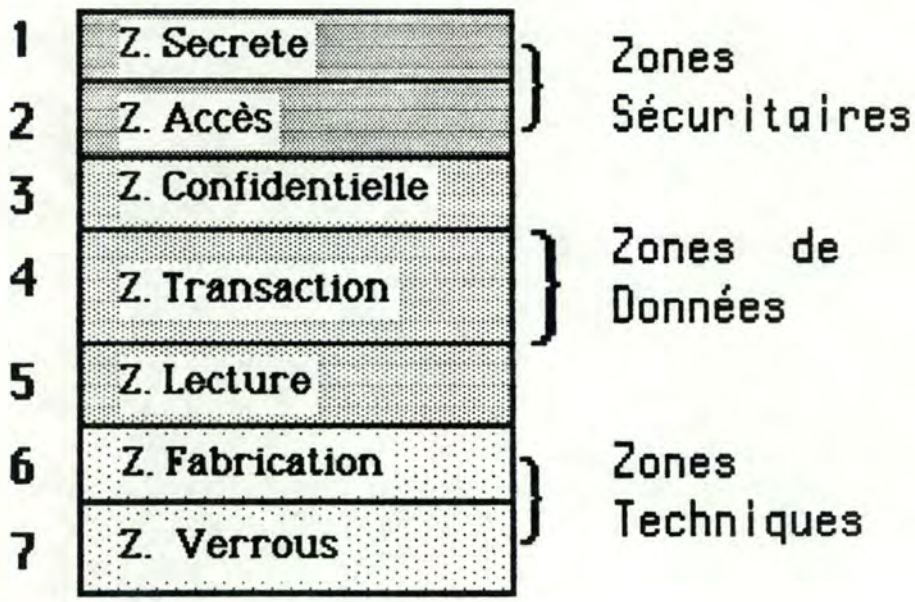


Fig. 3.5 : Les zones de la mémoire PROM

Les zones se répartissent en trois classes :

- les zones sécuritaires contenant des informations utiles au microprocesseur pour gérer l'accès à la mémoire PROM.

- les zones de données contenant les informations propres à l'application et au porteur de la carte .

- les zones techniques contenant des informations nécessaires à la gestion de la mémoire PROM.

Détaillons chacune des zones :

a) La zone secrète

La zone secrète ou zone des clés est une zone hautement protégée. En effet seul le microprocesseur peut en lire le contenu qui est une liste de clés :

- clé émetteur primaire,
- clé emetteur secondaire,
- clé porteur de type I,
- clé porteur de type II,
- clé interne
- clé de fabrication.

Il est important pour bien comprendre le rôle de ces clés de situer les utilisateurs de la carte :

- L' **émetteur** est un organisme qui propose un service à des clients.

  Il peut exister deux émetteurs pour une même carte. Dans ce cas, les deux émetteurs se partagent la mémoire disponible.

  Chaque émetteur doit pouvoir fixer les modalités d'utilisation de la carte (plafonds périodiques, ...) pour le service qu'il propose et il doit être le seul à pouvoir fixer ces modalités. Dans ce but , chaque émetteur dispose d'une clé qui lui est propre : **clé émetteur primaire, clé émetteur secondaire.**

- Le **porteur** est la personne qui utilise la carte pour un service proposé par un émetteur. Il possède une clé confidentielle qui l'identifie, appelée clé porteur.

  Initialement, c'est l'émetteur qui choisit la clé porteur , mais celle-ci peut être modifiée par le porteur. Il peut donc exister deux clés porteur pour une même carte : **clé porteur type I, type II** (à tout instant, une seule clé porteur est opérationnelle).

Voyons maintenant la fonction de chaque clé :

- clé émetteur (primaire ou secondaire) : elle permet à l'émetteur de fixer les modalités d'utilisation de la carte pour le service qu'il propose. Cette clé permet également l'accès aux informations protégées.

- clé porteur : elle permet au porteur de faire usage des droits fixés par l'émetteur (réaliser un achat chez le commerçant par exemple) et d'avoir accès aux informations protégées.

- clé interne : elle est utilisée comme paramètre de la fonction logico-mathématique de la carte et permet notamment l'authentification de la carte ainsi que la certification.

- clé de fabrication : elle est utilisée pour protéger l'accès à la mémoire PROM avant la mise en service de la carte. L'intérêt de cette clé sera mis en évidence au point 3.4 (Cycle de vie de la carte).

b) La zone d'accès

La zone d'accès est utilisée par le microprocesseur afin de mémoriser toutes les tentatives d'accès (correctes ou non) à une zone protégée c'est-à-dire nécessitant une présentation de clé.

La présentation d'un certain nombre de clés fausses entraîne le blocage de la carte (3 clés porteur fausses, 1 clé émetteur fausse). Lorsqu'une carte est bloquée, certaines opérations deviennent irréalisables (par exemple, l'accès à des informations protégées est impossible).

Une carte peut être débloquée (recyclée) après présentation des clés porteur et émetteur correctes .

La figure 3.6 illustre les états possibles d'une carte .

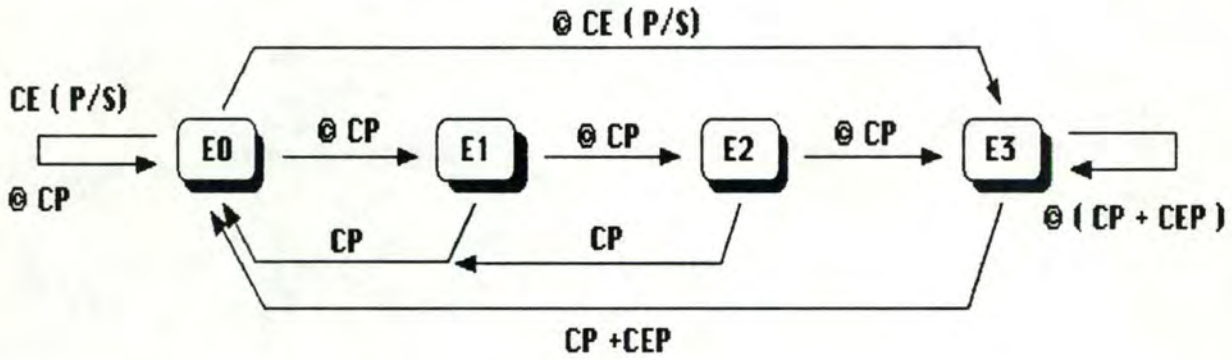


Fig. 3.6 : Les états de la carte

Les états sont les suivants :

E0 : état non bloqué, fonctionnement normal
E1 : état après 1 erreur
E2 : état après 2 erreurs
E3 : état bloqué

Les présentations de clés possibles sont les suivantes :

CP	: présentation d'une clé porteur correcte
CE	: présentation d'une clé émetteur correcte
⊗CP	: présentation d'une clé porteur incorrecte
⊗CE	: présentation d'une clé émetteur incorrecte
CP + CEP	: recyclage avec de bonnes clés porteur et émetteur primaire
⊗(CP + CEP)	: recyclage avec de mauvaises clés porteur ou émetteur primaire

N.B. P/S : primaire ou secondaire.

c) La zone confidentielle

La zone confidentielle contient des informations confidentielles propres à l'application. Celles-ci sont non évolutives.

d) La zone de transactions

La zone de transactions contient les autres informations propres à l'application et inscrites dans la carte à partir de sa mise en service. C'est en principe la plus grande zone de la mémoire PROM .

Y sont inscrits, par exemple, la liste des transactions effectuées par le porteur, les blocs de bits de prépaiement, ou encore les paramètres de connexion automatique à des services proposés sur des serveurs tels que le minitel.

e) La zone de lecture

La zone de lecture est semblable à la zone confidentielle. Cependant les informations qui y sont stockées (n° de compte du porteur, adresse ...), sont accessibles à toute personne. Leur lecture ne nécessite pas de présentation de clé.

f) La zone de fabrication

La zone de fabrication contient :

- les informations relatives à la taille des zones et à leur emplacement,
- les informations techniques relatives à la puce (n° de série , ...).

g) La zone des verrous

La zone des verrous signale la phase du cycle de vie dans laquelle la carte se trouve. Les différentes phases de vie seront détaillées au point 3.4. (La vie de la carte).

Sur la figure 3.7, nous trouvons le **contenu des 7 zones** :

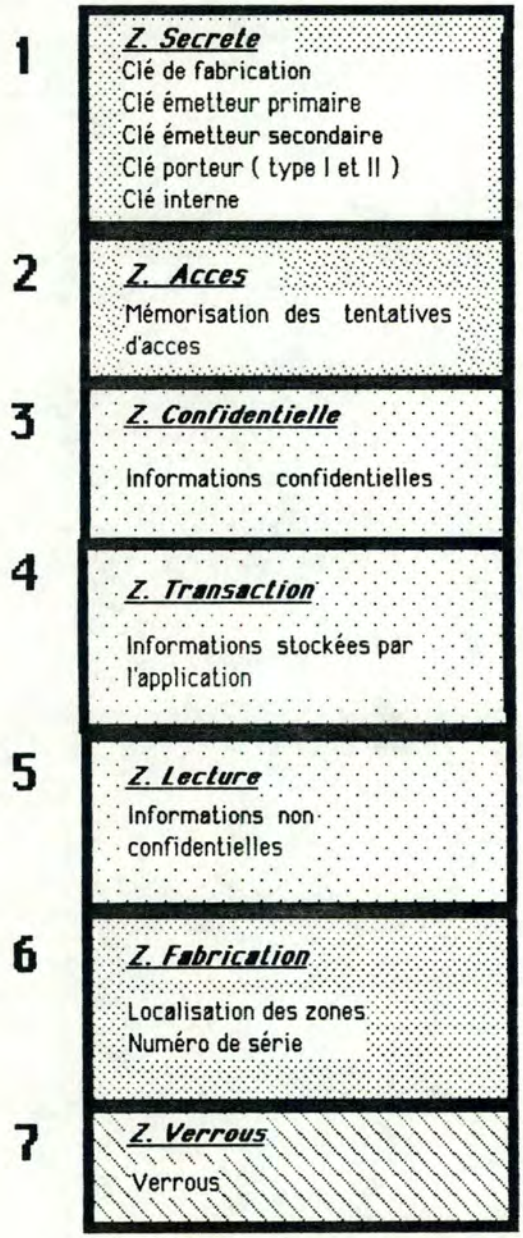


Fig. 3.7 : Le contenu des 7 zones

3.2.1.2. LES MOTS

Un mot compte 32 bits (fig. 3.8).

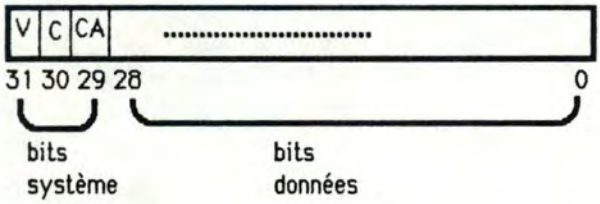


Fig. 3.8 : Le contenu d'un mot mémoire

Les 29 bits de poids faible sont les bits de données.
Les 3 bits de poids fort sont des bits systèmes.

Les bits C et CA spécifient le type d'utilisateur qui a écrit le mot (fig. 3.9).

Ces bits permettent, par exemple, au programme d'application de vérifier que les modalités d'utilisation de la carte pour un service donné ont bien été écrites par l'émetteur concerné ou que les transactions ont été réalisées par le porteur.

C	CA	
1	1	Mot écrit par l'émetteur primaire
1	0	Mot écrit par l'émetteur secondaire
0	X	Mot écrit par le porteur

Fig. 3.9 : Les bits C et CA

Le bit V permet de signaler qu'un mot est valide ou non.

Pour qu'un mot soit validé, la clé correspondant aux bits C et CA doit avoir été été présentée. Par exemple, un mot ayant 1 et 1 pour valeurs respectives des bits C et CA ne sera validé que si l'utilisateur a présenté au préalable la clé émetteur primaire.

Un mot validé ne peut plus jamais être modifié.

3.2.2. FONCTION LOGICO-MATHEMATIQUE .

Nous avons déjà signalé qu'en matière de sécurité de l'information, il est nécessaire de protéger les informations lors de leur communication ou de leur stockage; d'où l'utilité des algorithmes de chiffrement et bien sûr de déchiffrement.

Les cryptosystèmes sont des algorithmes qui ont pour but de transformer une information afin qu'elle devienne inintelligible et donc inutile pour ceux à qui elle n'est pas destinée.

On distingue trois grandes catégories de cryptosytèmes :

- Algorithme secret à clé secrète : l'algorithme doit être conservé secret ainsi que la clé de chiffrement.

- Algorithme public à clé secrète : l'algorithme peut être ici connu de tous. Seule la clé doit rester secrète .

- Algorithme public à clé révélée : l'algorithme, ainsi que la clé de chiffrement sont divulgués. Seule la clé de déchiffrement reste secrète. L'émetteur peut donc transmettre un message chiffré au récepteur qui, seul, possède la clé de déchiffrement. La clé de chiffrement (publique) est donc différente de celle de déchiffrement (secrète).

La fonction logico-mathématique de la carte est un cryptosystème développé par BULL CP8, couramment appelé algorithme Télépass. Télépass se situe dans la catégorie des algorithmes secrets à clé secrète. Il est implémenté sur la carte à microprocesseur et c'est la clé interne de celle-ci qui joue le rôle de clé secrète dans l'algorithme Télépass. A aucun moment, l'algorithme ou sa clé secrète n'est accessible de l'extérieur .

La fonction est de la forme suivante :

$$R = F (E , S , \text{adr carte} , (\text{adr carte}))$$

- **R** est le résultat de la fonction, appelé aussi certificat.

- **E** est une information fournie par l'extérieur appelée message d'entrée.

- **S** est la clé interne de la carte. Etant donné que cette information est secrète, elle garantit à R un caractère inimitable.

- **adr carte** est l'adresse d'un mot situé en mémoire PROM de la carte.

- **(adr carte)** est le contenu du mot dont l'adresse est adr carte (mot d'une mémoire protégée ou non). Ce mot est généralement appelé paramètre interne de la carte .

Le schéma de la figure 3.10 illustre cette fonction Télépass .

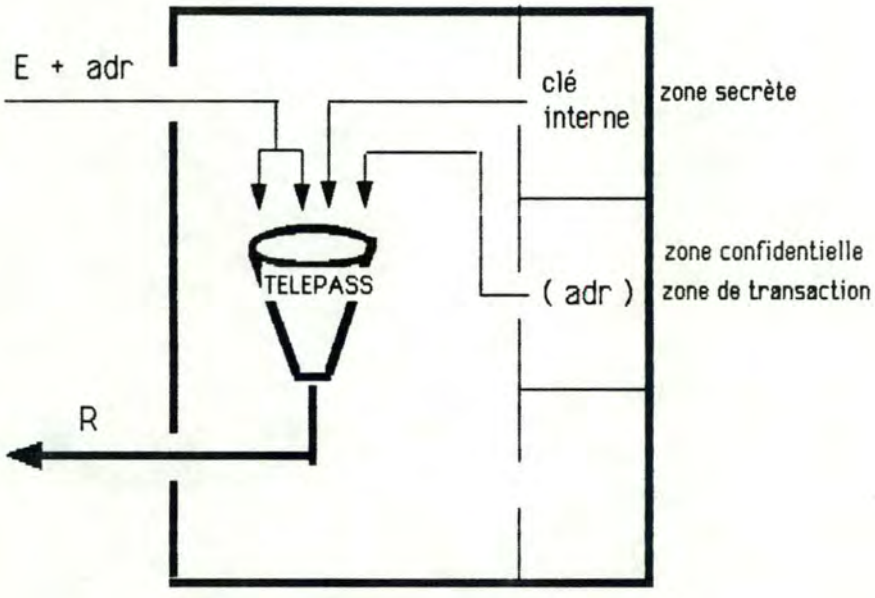


Fig. 3.10 : Schéma de Télépass

Cette fonction F n'est pas réversible. En d'autres termes, il est impossible de déduire de la forme de F une des trois formes ci-dessous :

F' tel que F' (S , R , adr , (adr)) ---> E

F'' tel que F'' (E , R , adr , (adr)) ---> S

F''' tel que F''' (S , R , E) ---> adr, (adr)

Cela signifie que Télépass est un algorithme de chiffrement pour lequel il n'existe pas d'algorithme de déchiffrement correspondant.

L' algorithme Télépass est utilisé pour sécuriser le dialogue entre la carte et un système informatique externe (terminal, ordinateur central, ...). Ce système informatique doit alors posséder un module sécuritaire comprenant un copie de l'algorithme et connaissant le secret S utilisé par la carte dans le dialogue.

3.2.3. GESTION DES ACCES A LA MEMOIRE PROM

Le microprocesseur est présent pour s'interposer en permanence entre le monde extérieur et la mémoire PROM. Cela signifie que toute action sur la mémoire PROM nécessite de la part du monde extérieur un ordre envoyé au microprocesseur .

Deux actions sur la mémoire sont autorisées : l'écriture et la lecture (la modification et l'effacement étant impossibles).

Les règles d'accessibilité pour le microprocesseur sont différentes des règles d'accessibilité pour les programmes d'application.

3.2.3.1 LECTURE ET ECRITURE DU MICROPROCESSEUR

Le microprocesseur peut accéder aux zones de la mémoire PROM de la façon suivante :

	ACCESSIBILITE	
	EN LECTURE	EN ECRITURE
ZONE SECRETE	oui	oui 1
ZONE D'ACCES	oui	oui
ZONE CONFIDENTIELLE	oui	non
ZONE TRANSACTION	oui	oui
ZONE DE LECTURE	oui	non
ZONE DE FABRICATION	oui	non
ZONE DES VERROUS	oui	oui

Fig. 3.11 : Droit d'accès du microprocesseur

1 : L'écriture dans la zone secrète ne peut se faire que dans le cas d'un changement de la clé porteur.

Comme la figure 3.11 le montre, seule l'écriture en zone confidentielle, en zone de lecture et en zone de fabrication est interdite au microprocesseur. La raison en est simple : après la mise en service de la carte, les trois zones citées précédemment sont déjà entièrement remplies.

3.2.3.2 LECTURE ET ECRITURE DE L'EXTERIEUR

Les programmes d'application disposent des possibilités suivantes :

	ACCESSIBILITE	
	EN LECTURE	EN ECRITURE
ZONE SECRETE	oui	oui 1
ZONE D'ACCES	oui 2	oui
ZONE CONFIDENTIELLE	oui 2	non
ZONE TRANSACTION	oui 23	oui 23
ZONE DE LECTURE	oui	non
ZONE DE FABRICATION	oui	non
ZONE DES VERROUS	oui	oui

Fig. 3.12 : Droit d'accès de l'extérieur

oui : La zone est totalement accessible sans présentation préalable de clé . Elle est dite accessible et libre .

non : La zone est totalement inaccessible .

1 : L'écriture dans la zone secrète ne peut se faire que pour un changement de clé porteur.

2 : La zone est accessible mais protégée c'est-à-dire que l'accès ne peut se faire qu'après présentation d'une clé porteur ou émetteur.

3 : La zone de transactions peut être protégée ou non selon le désir de l'émetteur .

Nous remarquons en comparant ces deux tableaux que le microprocesseur peut réaliser deux actions interdites aux programmes d'application :

- lecture dans la zone secrète : afin de pouvoir comparer la clé présentée par l'extérieur avec la clé secrète correspondante, ou pour utiliser la clé interne dans la fonction TELEPASS.

- écriture dans la zone d'accès : afin de comptabiliser les tentatives d'accès de l'extérieur .

3.2.4. JEU D INSTRUCTIONS DU MICROPROCESSEUR .

Les instructions sont de trois types :

- instructions d'initialisation,
- instructions basées sur l'algorithme Télépass,
- instructions simples.

Les instructions d'initialisation et les instructions simples renvoient généralement deux octets qui signalent comment l'instruction s'est déroulée et qui décrivent l'état de la carte à ce moment.

3.2.4.1. INSTRUCTIONS D'INITIALISATION

- La seule instruction de cette catégorie est la **remise à zéro**. Son rôle est de préparer le dialogue, c'est-à-dire initialiser la carte puis renvoyer au lecteur-encodeur de cartes les paramètres décrivant la carte insérée (conformément aux normes ISO).

3.2.4.2 INSTRUCTIONS BASEES SUR L'ALGORITHME TELEPASS

- Activation de la **fonction Telepass** : cette instruction calcule un certificat (R) au départ des arguments qui lui sont fournis (message d'entrée (E) , adresse du paramètre interne (adr) et des informations qui se trouvent dans la carte (clé interne (S) et paramètre interne (mot pointé par adr)).

L'adresse du paramètre interne doit être dans une zone accessible en lecture. Si cette zone est protégée en lecture, il est nécessaire d'avoir préalablement présenté et validé la bonne clé en lecture.

3.2.4.3. INSTRUCTIONS SIMPLES

- **Présentation d'une clé** : la présentation d'une clé est réalisée par trois ordres différents en fonction de la clé à présenter :

  - présentation d'une clé porteur ou d'une clé de fabrication,
  - présentation d'une clé émetteur primaire,
  - présentation d'une clé émetteur secondaire.

Ces ordres ont pour but de fournir une clé à la carte. La carte peut alors comparer la clé reçue avec celle présente en zone secrète. Le résultat de cette comparaison n'est pas fourni à l'extérieur .

- **Validation de clé en lecture** : cette instruction réalise la mise à jour de la zone d'accès à la suite d'une présentation de clé. Cette instruction doit être exécutée avant toute lecture et pour tout recyclage.

- **Lecture** : cette instruction a pour but de lire un nombre donné d'octets à partir d'une adresse donnée.

- **Ecriture d'un mot (32 bits)** : cette instruction réalise l'écriture d'un mot dans une zone protégée ou non en écriture . On ne peut écrire sur un mot validé, mais on peut réécrire sur un mot déjà utilisé mais non validé (par exemple pour écrire un seul bit).

- **Validation en écriture** : cet ordre réalise la validation d'un mot pour autant que les conditions de validation (définies au point 3.2.1.2.) soient remplies.

- **Lecture du résultat** : cette instruction retourne au monde extérieur le résultat R calculé par l'instruction d'activation de la fonction Télépass.

- **Ecriture des verrous** : cette instruction marque la fin d'une des phases de la vie de la carte en positionnant le verrou correspondant. Elle est aussi utilisée afin d'invalider une carte (l'invalidation sera définie au point 3.4.1.5).

- **Ecriture d'un mot de test** : pour mémoire.

3.2.5. GESTION DES ECHANGES ENTRE LA CARTE ET LE LECTEUR ENCODEUR

Le protocole adopté pour la gestion des échanges est conforme aux normes ISO sur les cartes à microprocesseur. Pour tout renseignement, se référer aux normes 7816/3 et 7816/3 Addendum1 en annexe 1.

3.3. LES PROGRAMMES D'APPLICATION

3.3.1. ENVIRONNEMENT D'EXECUTION

Les services accessibles au moyen de la carte à microprocesseur sont informatisés. Il doit donc exister des programmes d'application réalisant la mise en oeuvre de ces services.

De toute évidence, la carte ne possède pas une mémoire suffisante pour permettre le stockage de ces programmes d'application. De plus, ces derniers réalisent souvent des fonctions complémentaires à la gestion de la carte (par exemple, la mise à jour du compte client après une transaction). Les programmes d'application sont donc stockés et exécutés sur un système externe. En fonction de l'application , le système externe peut être :

- le lecteur-encodeur qui reçoit la carte (ces appareils sont généralement dotés d'un processeur et d'une mémoire d'environ 32 Koctets),
- un micro-ordinateur relié directement à un lecteur-encodeur,
- un gros système relié à une multitude de lecteurs-encodeurs via un réseau de télécommunications.

La figure 3.13 illustre le système externe constitué d'un micro-ordinateur et d'un lecteur-encodeur.

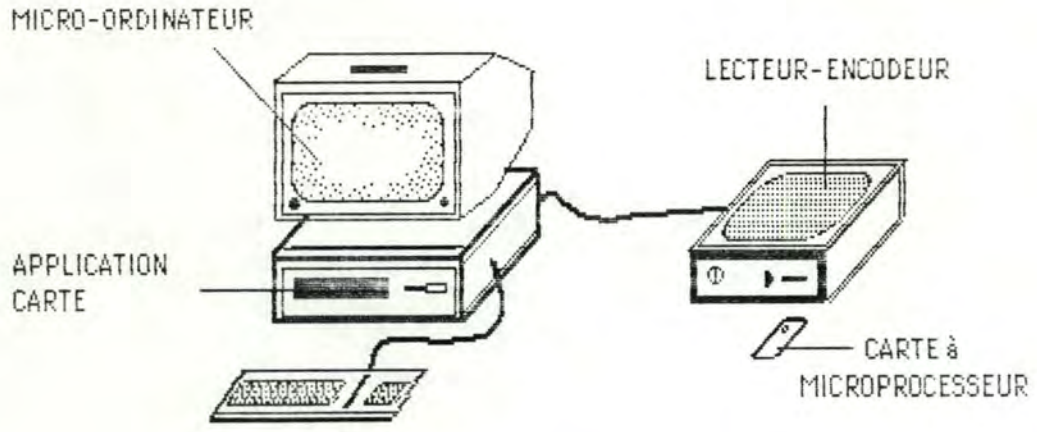


Fig. 3.13 : Un système externe

3.3.2. PRINCIPE DES COMMUNICATIONS AVEC LA CARTE

Pour les programmes d'application, la carte peut être perçue comme un moyen de stockage, au même titre que les disques ou les bandes magnétiques, et le microprocesseur peut être considéré comme un processeur d'entrées/sorties. Toutefois, à cause de la spécificité du support, les langages de programmation ne disposent pas de primitives standardisées permettant une manipulation aisée de la carte à microprocesseur.

Pour assurer la gestion de la carte, les programmes d'application ne peuvent utiliser que les seules instructions directement exécutables par le microprocesseur (cfr 3.2.4. Jeu d'instructions du microprocesseur).

En conséquence, les communications entre la carte et les systèmes externes sont constituées exclusivement de flots d'instructions appartenant au jeu d'instructions du microprocesseur et de leurs valeurs de retour.

3.3.3. EXEMPLES TYPIQUES DE COMMUNICATION AVEC LA CARTE

Les exemples les plus typiques de communication entre la carte et les programmes d'application - que nous allons détailler ici - sont les opérations de lecture et d'écriture. Nous verrons d'autres exemples de communication dans le chapitre 4 lorsque nous aborderons la réalisation des fonctions de sécurité.

Les opérations de lecture et d'écriture diffèrent lorsqu'elles sont réalisées sur des zones libres ou sur des zones protégées.

a) Lecture dans une zone libre

Pour réaliser une lecture en zone libre, il suffit d'utiliser l' **ordre de lecture** avec ses paramètres (adresse de début de lecture et nombre d'octets à lire) .

Les zones libres en lecture sont la zone de lecture, la zone de fabrication, la zone des verrous et, éventuellement, la zone de transactions.

b) Ecriture dans une zone libre

L'écriture dans une zone libre comporte deux étapes :

- L'envoi à la carte d'un **ordre d'écriture** et de ses paramètres (adresse de l'écriture et mot à écrire).

- L'envoi d'un ordre de **validation d'écriture** avec, comme paramètre, l'adresse du mot à valider.

Les zones libres en écriture sont la zone des verrous et, éventuellement, la zone de transactions.

Les lectures et les écritures en zone libre sont illustrées à la figure 3.14 :

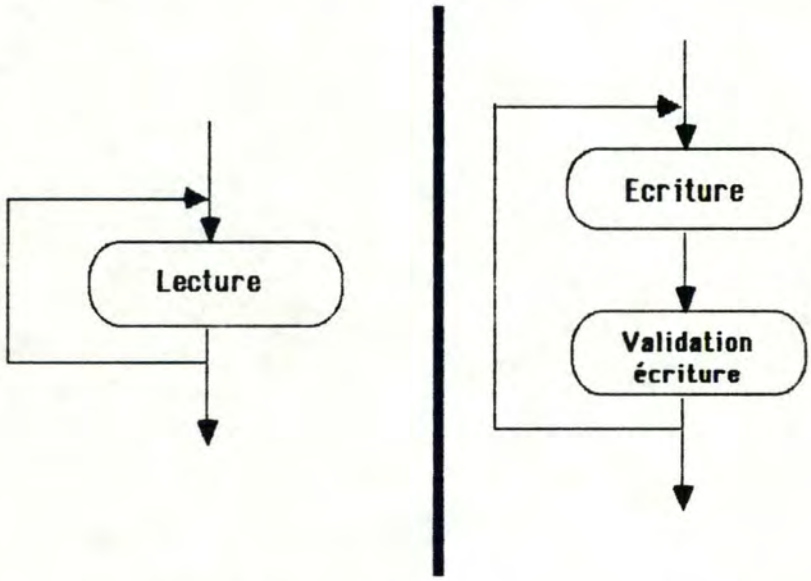


Fig. 3.14 : Scénario de lecture et d'écriture dans une zone non protégée

c) Lecture dans une zone protégée

Une lecture dans une zone protégée comporte trois étapes :

- La **présentation d'une clé** à la carte : soit une clé porteur, soit une clé émetteur.

- Une demande de **validation de cette clé en lecture**. Si la clé est correcte, toutes les lectures dans les zones protégées par la clé concernée seront autorisées. Dans le cas contraire, la carte comptabilise cette mauvaise présentation de clé dans la zone d'accès .

- L'envoi vers la carte d'un **ordre de lecture** et de ses paramètres .

Les zones protégées en lecture sont la zone d'accès, la zone confidentielle et, éventuellement, la zone de transactions.

d) Ecriture dans une zone protégée

Pour réaliser l'écriture dans une zone protégée , il est aussi nécessaire de passer par trois phases qui sont sensiblement les mêmes que celles vues pour la lecture protégée :

- La **présentation d'une clé** à la carte.

- L'envoi vers la carte d'un **ordre d'écriture** et de ses paramètres (adresse d'écriture et mot à écrire) .

- Une demande de **validation d'écriture** pour le mot qui vient d'être écrit.

Contrairement à la validation en lecture qui a lieu avant la lecture, la validation en écriture doit avoir lieu après l'écriture afin de pouvoir vérifier que les bits C et CA correspondent à la clé présentée.

La seule zone qui peut être protégée en écriture est la zone de transactions.

Les lectures et les écritures dans une zone protégée sont illustrées à la figure 3.15 :

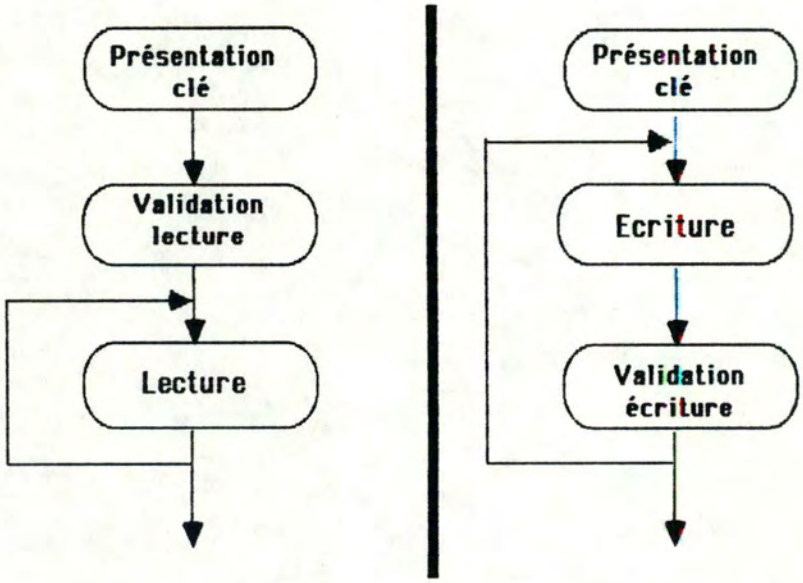


Fig. 3.15 : Scénario de lecture et d'écriture dans une zone protégée

3.4 CYCLE DE VIE DE LA CARTE

3.4.1. LES PHASES DE LA VIE D'UNE CARTE

La vie de la carte se schématise de la façon suivante :

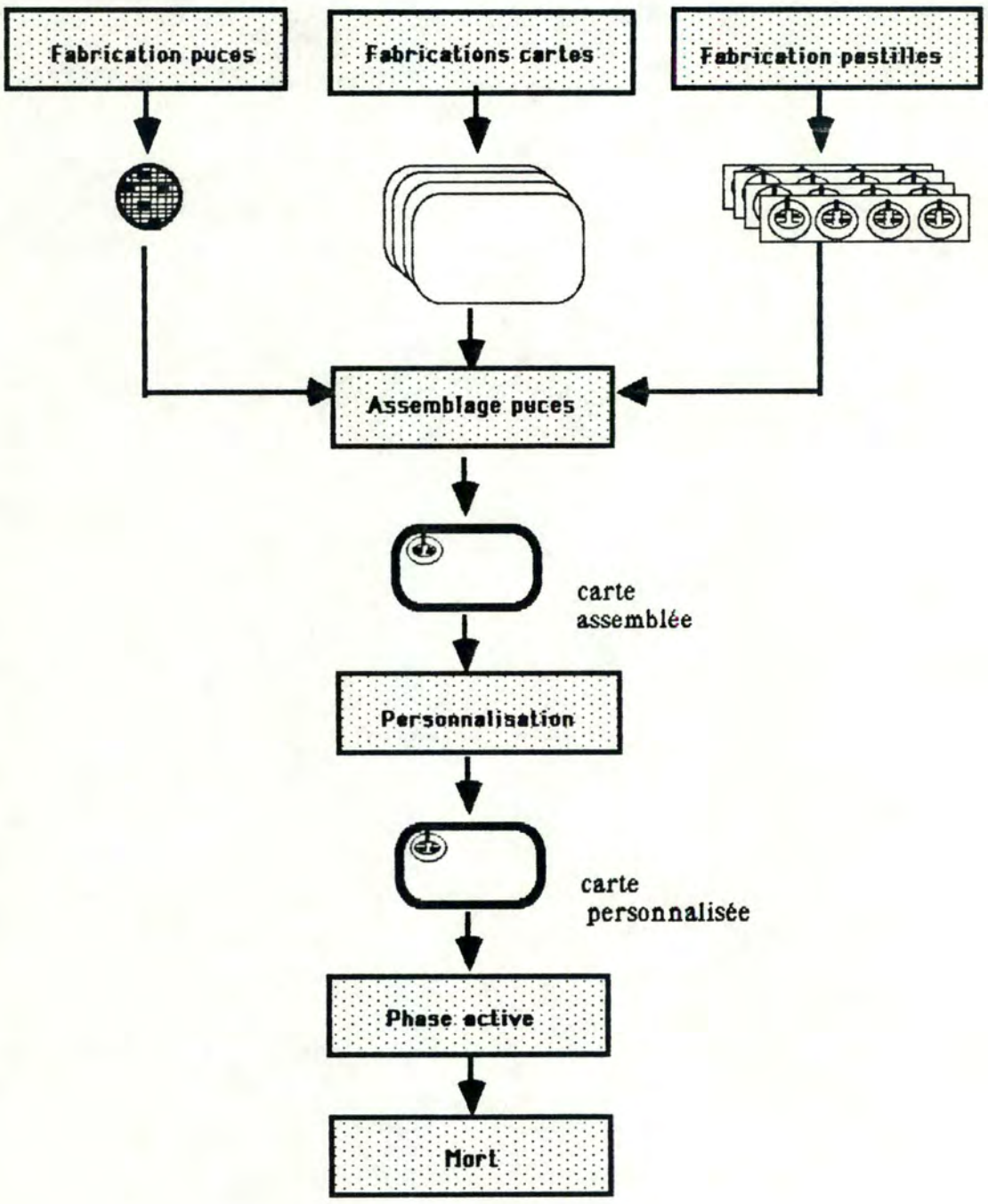


Fig. 3.16 : Vie de la carte

La vie de la carte englobe la fabrication de ses différents composants, leur assemblage, la personnalisation (c'est-à-dire l'initialisation des mémoires), la vie active (c'est-à-dire son utilisation pour des services divers) et se termine par la mort de la carte.

3.4.1.1. LA PHASE DE FABRICATION

La fabrication d'une carte comprend celle des puces, celle des pastilles et celle des cartes plastiques. Voyons de plus près la fabrication des puces. Durant cette phase :

- Le constructeur fabrique le microprocesseur, la ROM, la RAM et la PROM.

- Il écrit le masque, c'est-à-dire les programmes de la ROM.

- Il crée, dans la mémoire PROM, la zone de fabrication et celle des verrous .

- Il inscrit le numéro d'identification de la carte dans la zone de fabrication et la clé de fabrication dans la zone secrète. Il est important de noter que chaque carte du lot de fabrication possède déjà une clé de fabrication unique.

- On positionne, dans la zone des verrous, le verrou LF. Cela indique que la phase de fabrication est terminée et que les règles d'accessibilité à la mémoire PROM sont les suivantes :

    - lecture : libre, sauf pour la clé de fabrication.
    - écriture : protégée (aucune écriture ne peut être tentée sans présentation préalable de la clé de fabrication).

Le lot de puces est ainsi fabriqué puis envoyé vers l'endroit d'assemblage.

3.4.1.2. LA PHASE D'ASSEMBLAGE

La phase d'assemblage comprend :

- le test des puces,
- l'encartage de la puce et de la pastille dans la carte plastique,
- l'écriture d'un numéro de série.

3.4.1.3. LA PHASE DE PERSONNALISATION

La personnalisation consiste à inscrire dans la mémoire PROM les informations caractéristiques des services proposés ainsi que les données afférentes au destinataire final de la carte (le porteur).

Plus précisément, la personnalisation comprend :

- l'écriture des informations définissant la taille, la localisation des différentes zones (secrète, accès, confidentielle, transactions, lecture),

- l'écriture du type de protection en lecture et en écriture de la zone de transactions,

- l'écriture dans les zones appropriées des informations suivantes :

    - données sur le futur porteur de la carte (identité bancaire, adresse),
    - données relatives au service offert par la carte,
    - les clés secrètes : celle du porteur , du (des) émetteur(s) et enfin la clé interne .

- le positionnement du verrou indiquant la terminaison de la phase de personnalisation (la clé de fabrication est alors désactivée tandis que les clés porteur, émetteur et interne sont activées).

Cette phase est réalisée par l'emetteur des cartes qui est le seul à connaître la clé de fabrication de chacune des cartes.

A partir de la fin de la personnalisation, les règles d'accessibilité à la mémoire PROM sont celles définies au point 3.2.3 (Gestion des accès à la mémoire PROM .

La carte est alors envoyée à chaque porteur. La clé porteur leur est communiquée par une autre voie.

Remarque : la personnalisation des cartes est réalisée par des programmes analogues aux programmes d'application.

3.4.1.4. LA PHASE ACTIVE

La phase active correspond à la phase d'utilisation de la carte par son porteur.

Les actions réalisables par le porteur dépendent des services en fonction desquels la carte a été personnalisée.

A titre indicatif, citons les actions les plus courantes :

- effectuer des transactions,
- visualiser la liste des transactions déjà réalisées,
- changer la valeur de la clé porteur (la première clé devient alors inactive),
- débloquer une carte.

3.4.1.5. LA MORT DE LA CARTE

La mort de la carte peut survenir dans quatre cas :

- saturation de la zone d'accès,
- saturation de la zone de transactions,
- invalidation de la carte par le programme d'application (par exemple lorsque la carte insérée figure sur une liste noire),
- auto-invalidation de la carte dès qu'une condition anormale d'utilisation est détectée (par exemple lorsque la mémoire PROM a été effacée).

Remarque : invalider une carte consiste à inhiber certaines instructions du microprocesseur. L'invalidation est réalisée par le positionnement d'un verrou dans la zone des verrous. Lors de toute mise sous tension de la carte , le verrou d'invalidation est testé. Si ce dernier est positionné, le microprocesseur refusera d'exécuter les instructions inhibées.

3.4.2. PROTECTION DE LA CARTE PENDANT LE CYCLE DE VIE

La carte est protégée contre une utilisation illégitime durant toutes les étapes du cycle de vie.

Les sites de fabrication, d'assemblage et de personnalisation étant souvent différents, la puce est transmise de l'usine de fabrication vers le porteur en passant par l'usine d'assemblage et par l'émetteur. Il faut donc se prémunir contre les vols de cartes et, le cas échéant, éviter que le voleur puisse ensuite utiliser les cartes volées tout à fait normalement .

C'est pourquoi, dès la fin de la fabrication de la puce, une clé est inscrite dans la carte **(clé de fabrication)**. Dès ce moment, on ne peut plus écrire sur la carte sans avoir au préalable présenté cette clé de fabrication. Il est donc impossible pour un fraudeur de personnaliser une carte volée après la fabrication.

Lorsqu'une carte a été personnalisée, la clé de fabrication est désactivée au profit des autres clés contenues dans la zone secrète (clé émetteur, clé porteur, clé interne). Ce sont alors ces nouvelles clés qui préviennent toute utilisation abusive de la carte.

CHAPITRE 4 :
EVALUATION DE LA CARTE A MICROPROCESSEUR

4.1. PROTECTION DE L'INFORMATION

- 4.1.1. Sécurité physique
- 4.1.2. Sécurité logique
  - 4.1.2.1. Implémentation des fonctions de sécurité
    - a. Authentification de l'utilisateur
    - b. Authentification de la carte
    - c. Certification
    - d. Signature électronique
    - e. Génération de clés de chiffrement
  - 4.1.2.2. Confidentialités des clés
    - a. Chiffrement
    - b. Innacessibilité
    - c. Diversification
    - d. Blocage

4.2. POLYVALENCE

4.3. ASPECT MULTI-SERVICES

- 4.3.1. Nombre de services existants
- 4.3.2. Indépendance des services

IV. EVALUATION DE LA CARTE A MICROPROCESSEUR

Dans ce chapitre, nous allons voir dans quelles mesures la carte décrite au chapitre trois permet d'atteindre les objectifs vus au chapitre deux (protection de l'information, polyvalence et aspect multi-services).

4.1. PROTECTION DE L'INFORMATION .

Avant de voir en détail la protection de l'information mise en oeuvre grâce à la carte, nous tenons à faire remarquer que le niveau de protection atteint est très élevé. Dans notre étude, nous n'avons relevé qu'une seule faiblesse que nous signalerons en temps opportun.

4.1.1 SECURITE PHYSIQUE

Les dispositifs de sécurité implémentés au niveau physique découlent de la **nature et de l'architecture de la puce.** Les protections sont les suivantes:

- La carte résiste aux champs électromagnétiques, aux rayons UV et aux rayons X inférieurs à 20000 rads.

- La technologie utilisée pour la fabrication des puces (technologie FAMOS) garantit que les mémoires ne sont lisibles que via le bus des données et celui des adresses.

- La structure monolithique de la puce empêche tout accès aux bus qui ne serait pas autorisé par le microprocesseur (à l'exception des accès via les plots de test).

- Les plots de tests qui permettent un accès direct aux bus internes et qui sont utilisés pour tester les mémoires avant la mise en service des cartes sont détruits à la fin de la phase de fabrication.

- L'effacement global de la mémoire EPROM, qui est théoriquement possible par rayons UV, est contrôlé par le microprocesseur. Ce dernier peut détecter un effacement de sa mémoire de stockage en testant la valeur de certains bits appelés témoins d'effacement. En cas d'effacement, le microprocesseur peut rendre la carte inutilisable.

- En cas d'utilisation anormale (chute de tension de l'écriture, ralentissement de l'horloge, ...), le microprocesseur se bloque.

- Les plans de fabrication de la puce sont secrets. Cela rend la duplication d'une carte très peu probable car cette duplication nécessiterait des compétences et des investissements considérables.

4.1.2. SECURITE LOGIQUE .

La sécurité logique repose sur les cinq fonctions de sécurité vues au chapitre deux. Toutefois, il ne suffit pas d'implémenter ces cinq fonctions pour garantir un haut niveau de sécurité logique. En effet, ces cinq fonctions sont basées sur l'utilisation de clés secrètes, et la divulgation de ces clés rendrait les fonctions de sécurité inutiles. Pour garantir un haut degré de sécurité logique, il faut donc :

- implémenter les fonctions de sécurité,
- assurer la confidentialité des clés.

4.1.2.1. IMPLEMENTATION DES FONCTIONS DE SECURITE

a) Authentification de l'utilisateur

La figure 4.1 illustre le principe de l'authentification de l'utilisateur.

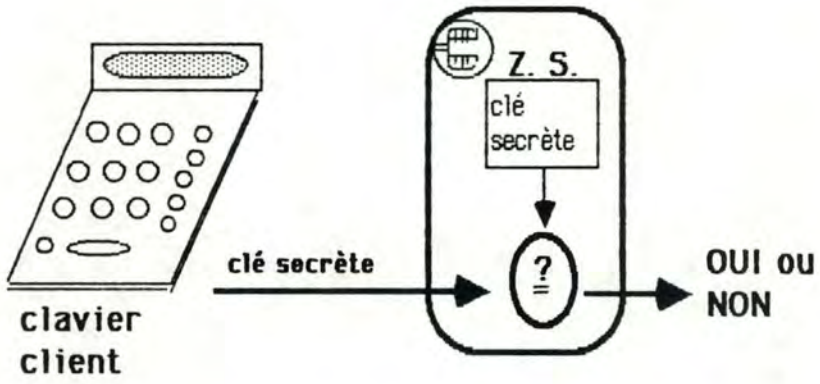


Fig 4.1 : L'authentification de l'utilisateur

Le principe d'authentification de l'utilisateur est simple : après avoir été saisie au clavier, la clé secrète (clé porteur, clé émetteur ou clé de fabrication) est envoyée à la carte qui la compare avec celle stockée dans la zone secrète.

Cette fonction est réalisée par l'instruction de **présentation de clé**.

b) Authentification de la carte

Lorsqu'un système externe veut s'assurer de l'authenticité d'une carte, il lui demande de présenter sa clé interne.

Rappelons qu'un système externe possède un module sécuritaire qui dispose de l'algorithme Télépass et qui connaît la clé interne de chaque carte avec laquelle le système externe est susceptible de dialoguer.

L'authentification de la carte est réalisée par le programme d'application du système externe de la manière suivante :

- Le programme d'application génère un nombre aléatoire E.
- Le programme d'application demande à la carte d'**activer la fonction Télépass** et lui fournit comme arguments le nombre aléatoire E et l'adresse ADR d'un mot situé dans la mémoire de stockage de la carte.
- Le programme d'application envoie à la carte l'ordre de **lecture du résultat** et reçoit en retour un certificat R.
- Le programme d'application réalise une **lecture** du mot situé à l'adresse ADR (parfois cette étape peut être évitée si le programme d'application connaît à priori le contenu de ce mot).

- Le programme d'application calcule un certificat R' en prenant comme paramètres :

  - le nombre aléatoire E,
  - l'adresse ADR,
  - le contenu (ADR) du mot situé à l'adresse ADR,
  - la clé interne de la carte à authentifier.

- Le programme d'application compare les certificats R et R'. S'ils concordent, cela signifie que la clé interne de la carte est correcte, et donc que la carte est authentique.

La figure 4.2 illustre le principe d'authentification d'une carte.

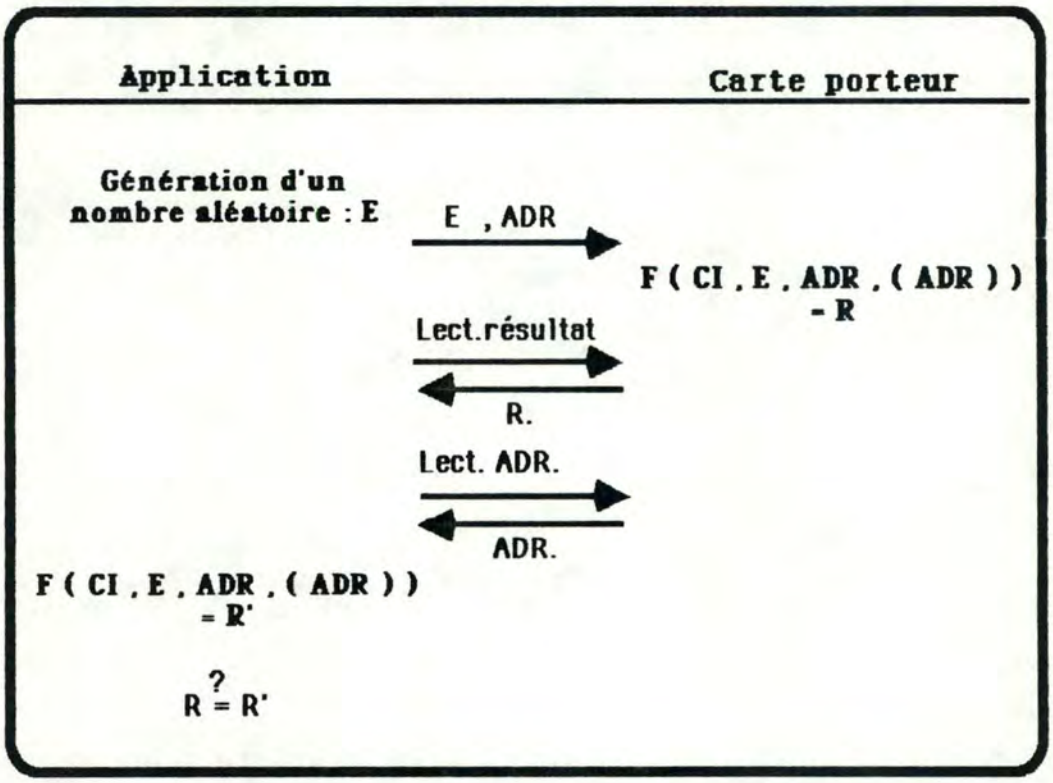


Fig 4.2 : L'authentification d'une carte

c) Certification

Rappelons que la certification consiste à fournir un certificat relatif à une information déterminée et que le certificat sert à garantir la présence de cette information dans la carte.

La certification est réalisée sur un ordre du programme d'application du système externe de la manière suivante :

- Le programme d'application génère un nombre aléatoire E.

- Le programme d'application demande à la carte d'**activer la fonction Télépass** et lui fournit comme arguments le nombre aléatoire E et l'adresse ADR du mot pour lequel le système externe veut un certificat.

- Le programme d'application envoie à la carte l'ordre de **lecture du résultat** et reçoit en retour un certificat R.

- Le programme d'application calcule un certificat R' en prenant comme paramètres :

    - le nombre aléatoire E,
    - l'adresse ADR,
    - le contenu (ADR) du mot situé à l'adresse ADR (ce mot doit être connu à priori par le programme d'application),
    - la clé interne de la carte.

- Le programme d'application compare les certificats R et R'. S'ils concordent, cela signifie que le mot situé à l'adresse ADR est effectivement celui attendu par le programme d'application.

La figure 4.3 illustre le principe de la certification.

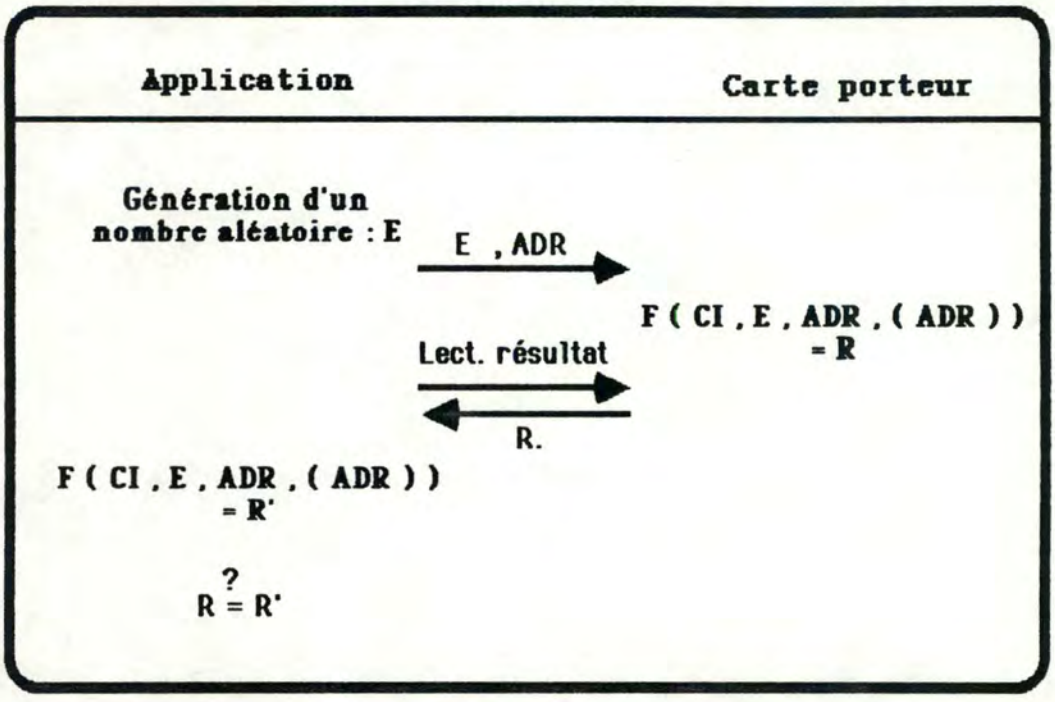


Fig 4.3 : La certification

Remarque : à première vue, on pourrait penser qu'une simple lecture du mot situé à l'adresse ADR est équivalente à une certification. Il n'en est rien! En effet, si la lecture permet de garantir l'existence de l'information, rien ne prouve que cette information se trouve sur une carte déterminée car la lecture ne fait pas référence à la clé interne de la carte.

d) Signature électronique

Rappelons que la signature électronique est utilisée pour vérifier l'authenticité et l'absence d'altérations d'un message au cours d'une transmission.

Pour cette fonction, le dialogue ne s'établit plus entre un programme d'application et une carte comme précédemment, mais entre un programme émetteur et un programme récepteur. Ces deux programmes doivent disposer respectivement d'une carte CP8 et d'un module sécuritaire.

Le principe de la signature électronique est le suivant :

- L'émetteur condense le message M à transmettre en une chaîne de six octets désignée par E.

- L'émetteur génère un certificat R en utilisant comme paramètres :

    - le message condensé E,
    - l'adresse ADR d'un mot situé dans la mémoire de stockage de la carte (le contenu de ce mot indique généralement l'identité de l'émetteur),
    - le contenu (ADR) du mot dont l'adresse est ADR,
    - la clé interne de la carte utilisée pour le dialogue.

- L'émetteur envoie au récepteur :

    - le message M,
    - le certificat R,
    - l'adresse ADR,
    - éventuellement le contenu (ADR) du mot situé à l'adresse ADR.

- Le récepteur condense le message reçu M' en une chaîne de six octets désignée par E'.

- Le récepteur génère un certificat R' en utilisant comme paramètres :

    - le message condensé E',
    - l'adresse ADR,
    - le contenu (ADR) du mot dont l'adresse est ADR,
    - la clé interne de la carte.

- Le récepteur compare les certificats R et R'. S'ils concordent, les messages M et M' sont identiques.

La figure 4.4 résume le principe de la signature électronique.

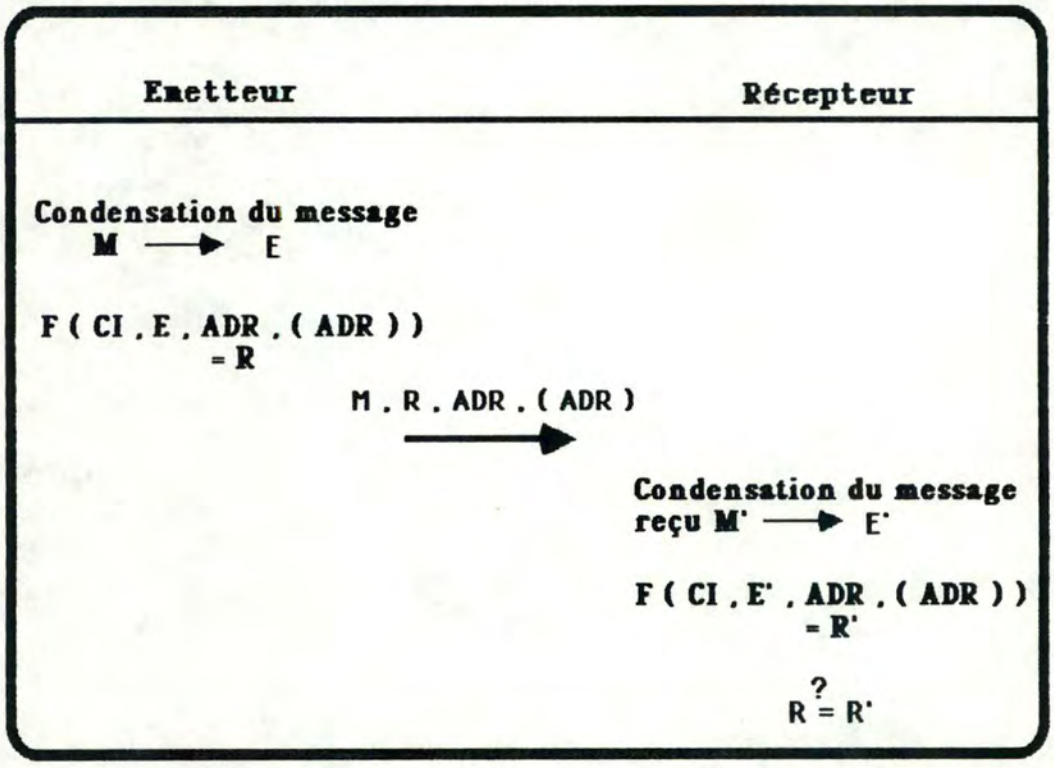


Fig 4.4 : La Signature électronique

Remarque : la signature électronique pourrait être réalisée de la même manière si l'émetteur disposait du module sécuritaire et si le récepteur disposait de la carte CP8.

e) Génération de clés de chiffrement

Pour réaliser la génération de clés de chiffrement, les programmes de chiffrement et de déchiffrement doivent disposer respectivement d'un module sécuritaire et d'une carte CP8.

Le principe de la génération de clés de chiffrement est le suivant :

- Le programme de chiffrement génère un nombre aléatoire E.

- Le programme de chiffrement envoie au programme de déchiffrement le nombre aléatoire E et l'adresse ADR d'un mot situé dans la mémoire de stockage de la carte.

- Le programme de chiffrement réalise une **lecture** du mot situé à l'adresse ADR (parfois cette étape peut être évitée si le programme de chiffrement connaît à priori le contenu de ce mot).

- Le programme de chiffrement calcule un certificat R en prenant comme paramètres :

    - le nombre aléatoire E,
    - l'adresse ADR,
    - le contenu (ADR) du mot situé à l'adresse ADR,
    - la clé interne de la carte.

- Le programme de déchiffrement calcule un certificat R' en prenant comme paramètres :

    - le nombre aléatoire E,
    - l'adresse ADR,
    - le contenu (ADR) du mot situé à l'adresse ADR,
    - la clé interne de la carte.

- Sauf erreur de transmission, les certificats R et R' sont identiques et constituent une clé de chiffrement.

La figure 4.5 illustre le principe de la génération de clés de chiffrement.

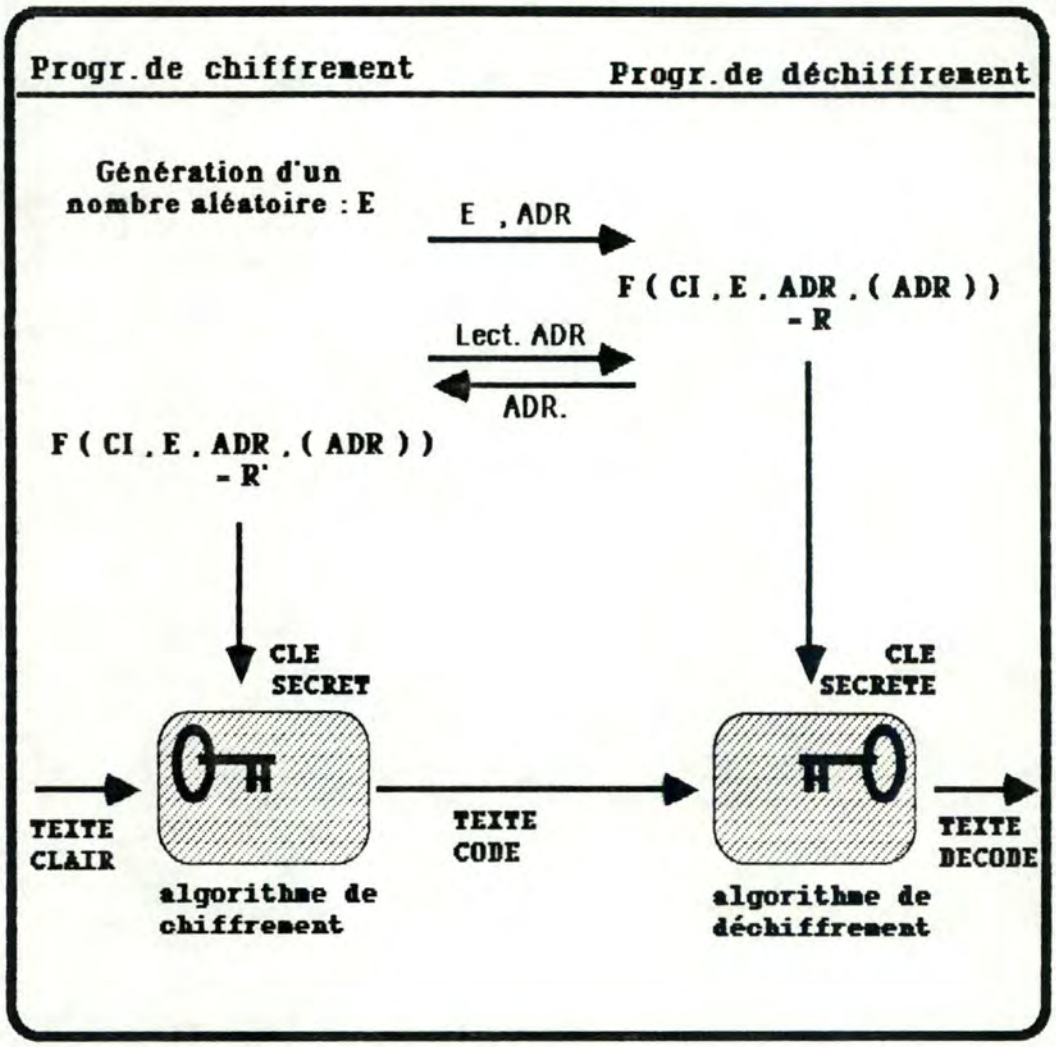


Fig 4.5 : Génération de clés de chiffrement

Remarque : la génération de clés de chiffrement serait également réalisable si le programme de chiffrement disposait de la carte CP8 et si le programme de déchiffrement disposait du module sécuritaire. Nous laissons au lecteur le soin d'imaginer les modifications que cela entraînerait dans le principe décrit ci-dessus.

4.1.2.2. CONFIDENTIALITES DES CLES

Le caractère confidentiel des clés est assuré par :

- le chiffrement,
- l'inaccessibilité,
- la diversification,
- le blocage.

a) Chiffrement

Le chiffrement sert à éviter l'interception des clés pendant un dialogue. Chaque fois qu'une clé doit circuler de la carte vers un système externe (par exemple pour une authentification de carte) - ou l'inverse - elle est remplacée par un certificat. Ce dernier prouve au récepteur que l'émetteur connaît la clé, mais la connaissance du certificat ne permet pas à un éventuel fraudeur de deviner la clé utilisée.

L'authentification de l'utilisateur fait exception à cette règle. Cette exception se justifie aisément :

- L'utilisateur ne dispose pas de l'algorithme Télépass. Il lui est donc impossible de générer un certificat.

- L'authentification de l'utilisateur ne nécessite pas de dialogue avec des systèmes distants. Les possibilités d'interception de la clé secrète sont donc fortement réduites.

b) Inaccessibilité

Puisque les clés ne peuvent pas être interceptées pendant le dialogue, le seul moyen de les connaître est de remonter à leur source c'est-à-dire la carte, le module sécuritaire ou l'utilisateur.

- **La carte**

Toutes les clés sont stockées dans des mots qui ne sont accessibles de l'extérieur ni en lecture, ni en écriture. Dès qu'elles sont écrites dans la carte, seul le microprocesseur peut encore y avoir accès :

- en lecture : uniquement pour un usage interne.
- en écriture : uniquement lors du changement de la clé porteur.

- **Le module sécuritaire**

Le module sécuritaire est une carte CP8 utilisée uniquement pour l'implémentation des fonctions de sécurité et pour la diversification des clés (cfr point c. Diversification). Les clés qui y sont stockées sont donc protégées de la même manière que sur une carte CP8 classique.

- **Les utilisateurs**

Les utilisateurs (porteur et émetteur) doivent veiller eux-mêmes à ne pas divulguer leur clé, soit oralement, soit en la présentant à un appareil piraté.

De toute évidence, la sécurité des clés détenues par les utilisateurs n'est pas absolue. C'est pourquoi, certains masques prévoient des clés supplémentaires (par exemple, une forme digitalisée des empreintes digitales ou de la signature).

c) Diversification

Nous avons dit à plusieurs reprises que l'accès physique à la mémoire de la carte était impossible. C'est exact actuellement, mais il est tout à fait plausible que certaines protections physiques de la puce puissent être outrepassées dans un avenir plus ou moins proche (selon les capitaux investis dans ce type de recherches). Il est donc inconcevable de se fier aveuglément au principe d'inaccessibilité des clés.

La diversification constitue un système de sécurité complémentaire destiné à éviter que la découverte des clés d'une carte puisse servir pour une fraude à grande échelle. Cet objectif est réalisé en fournissant à chaque carte des clés qui lui sont propres.

Toutefois, un problème se pose si chaque carte dispose de clés propres. Comment un module sécuritaire - dont la capacité de mémorisation est limitée - peut-il connaître la clé interne de chacune des cartes avec lesquelles il est susceptible de dialoguer? De même, comment un émetteur peut-il connaître la clé émetteur des cartes qu'il distribue sans avoir recours à de longues listes de clés dont la confidentialité ne peut être que difficilement garantie?

Pour résoudre ce problème, le système de diversification permet de remplacer la mémorisation de clés par le calcul de clés. Le principe de la diversification (fig. 4.6.) est le suivant :

Lors de la personnalisation d'une carte - appelée couramment carte fille, les clés stockées dans la zone secrète sont choisies de manière à pouvoir être recalculées à tout moment.

A partir de la clé interne du module sécuritaire - également appelé carte mère, du numéro de série de la carte fille et d'un paramètre interne au module sécuritaire, l'émetteur calcule un certificat. Ce certificat constitue une clé de la carte fille.

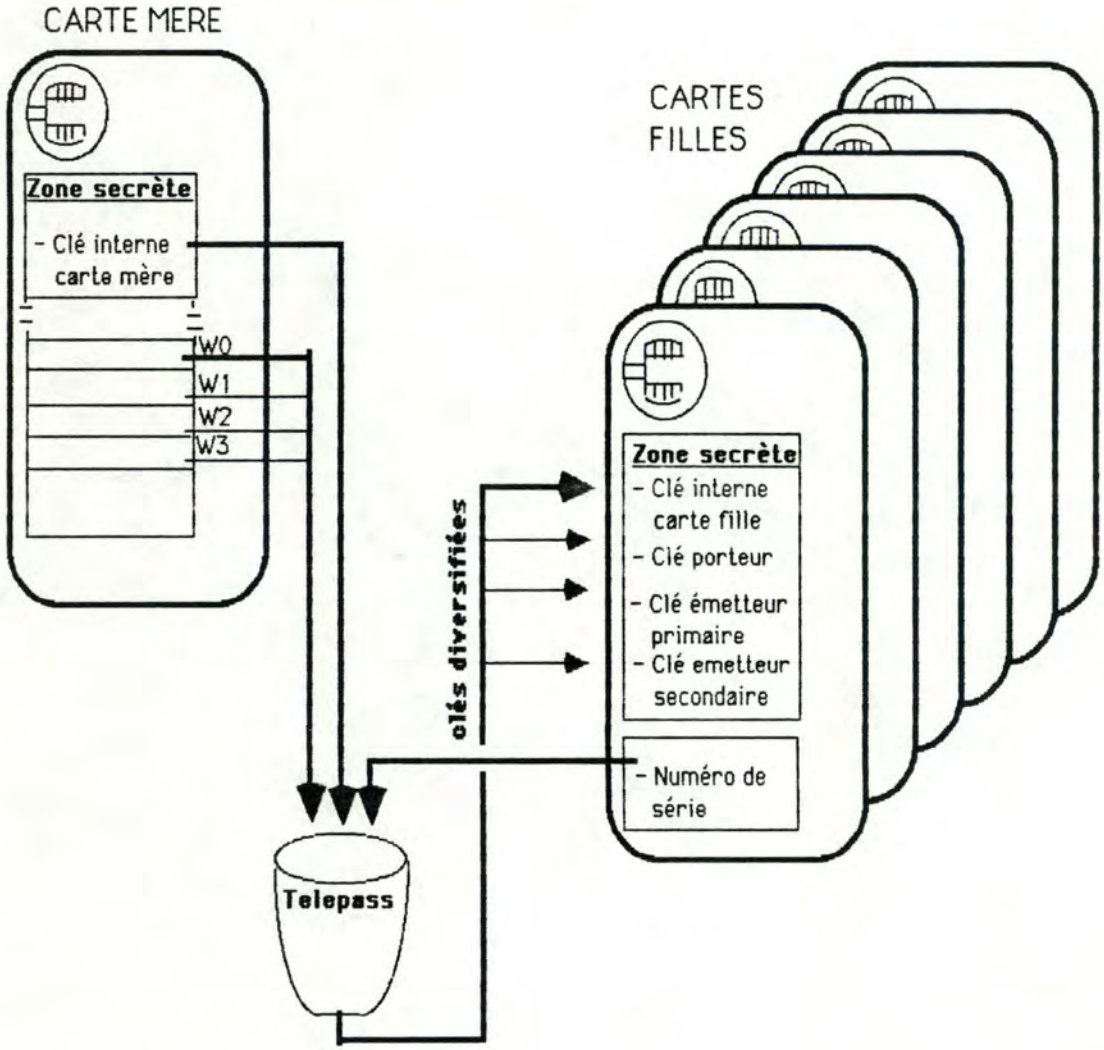


Fig. 4.6 : Principe de la diversification

Comme la figure 4.6 nous le montre, quatre des six clés de la zone secrète sont diversifiées au moyen d'une carte mère. Tout programme d'application disposant d'un module sécuritaire adéquat peut donc retrouver à tout moment la valeur de chacune de ces clés. Il lui suffit pour cela de lire le numéro de série de la carte avec laquelle il dialogue et, ensuite, de calculer un certificat en utilisant le paramètre interne correspondant à la clé qu'il veut connaître.

L'utilisation du numéro de série de la carte fille comme paramètre de diversification permet de diversifier les clés d'un grand nombre de cartes au départ d'un seul module sécuritaire.

La diversification ne sert pas uniquement pour le calcul des clés émetteur, porteur et interne. Elle sert également pour le calcul de la clé de fabrication. Dans ce cas, le principe de diversification est identique mais le module sécuritaire utilisé est appelé carte lot.

La figure 4.7 résume l'emploi de la diversification tout au long de la vie de la carte.

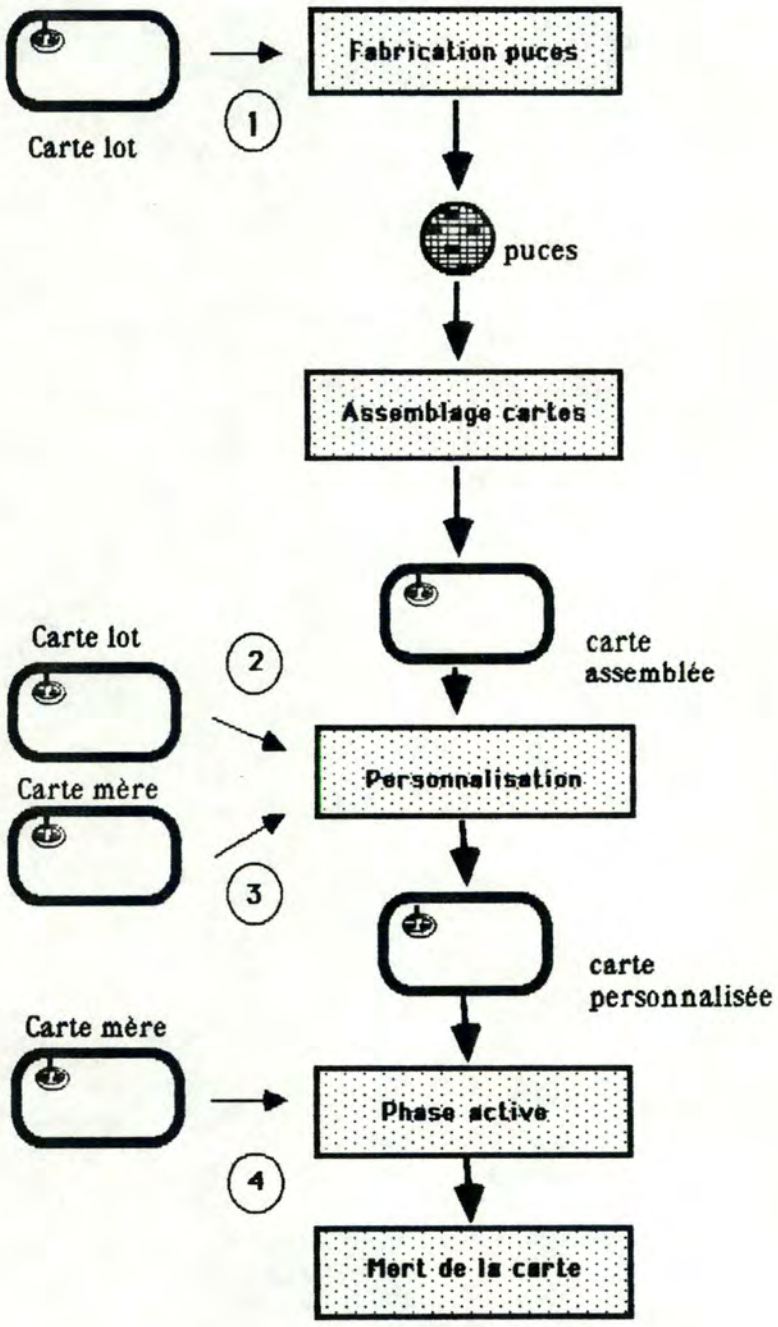


Fig. 4.7 : Diversification et vie de la carte

1. Le constructeur génère la clé de fabrication et écrit celle-ci dans la carte.

2. L'émetteur calcule la clé de fabrication et la présente à la carte afin de pouvoir réaliser les écritures requises pour la personnalisation.

3. L'émetteur génère les clés de la zone secrète (hormis la clé porteur de type deux) et les écrit dans la carte.

4. L'émetteur calcule la clé qui doit être utilisée dans une fonction de sécurité. Cette étape, contrairement aux trois premières qui n'ont lieu qu'une seule fois, a lieu chaque fois qu'une fonction de sécurité impliquant une clé diversifiée (autre que la clé de fabrication) doit être exécutée.

Comme nous le voyons sur la figure 4.7, les modules sécuritaires sont utilisés très fréquemment. En particulier, la carte mère est nécessaire pratiquement lors de chaque utilisation d'une carte CP8 par son porteur. Il est donc indispensable, dans le cadre d'applications décentralisées (systèmes off-line), de disposer de nombreuses copies de la carte mère.

C'est là que réside la faiblesse de la philosophie CP8. En effet, l'interception d'une seule de ces copies rendrait les fonctions de sécurité inopérantes contre le possesseur de cette copie. Le concepteur d'applications doit être bien conscient de cette possibilité afin d'être en mesure de prévoir des systèmes de sécurité complémentaires.

d) Blocage

Le blocage consiste à inhiber certaines instructions du microprocesseur s'il y a présomption de fraude. Il a pour but d'empêcher toute recherche systématique des clés d'authentification de l'utilisateur par essais et erreurs.

La carte s'autobloque lorsqu'elle s'aperçoit qu'une personne non habilitée essaie de l'utiliser. En d'autres termes, la carte se bloque lorsque l'utilisateur ne réussit pas à passer l'authentification avec succès. Le blocage est rendu possible grâce à la zone d'accès qui enregistre le résultat de chaque présentation de clé.

Remarques : - Il ne faut pas confondre le blocage et l'invalidation. Le blocage n'est pas définitif et est entièrement géré par la carte. Par contre, l'invalidation est définitive (cfr 3.4.1.5. Mort de la carte) et peut avoir lieu soit sur ordre du programme d'application, soit sur un ordre du microprocesseur.

- Les présentations de clés de fabrication ne sont pas enregistrées dans la zone d'accès, mais dans le premier mot de la mémoire PROM. Le blocage survient lorsque trois clés de fabrication fausses ont été présentées successivement. Une carte ainsi bloquée ne peut pas être débloquée.

4.2. POLYVALENCE

Au niveau de la polyvalence, nous pouvons dire que la carte CP8 est un produit tout à fait satisfaisant. La grande polyvalence de la carte découle principalement de deux caractéristiques.

Tout d'abord, l'émetteur dispose d'une certaine liberté lors de la personnalisation de la carte, ce qui lui permet de configurer la carte en fonction de l'application. Plus précisément, il peut choisir :

- la taille respective des différentes zones (certaines zones peuvent même ne pas exister),

- le type de protection de la zone de transactions (cette zone peut être libre, protégée en lecture, protégée en écriture ou protégée en lecture et en écriture).

Par exemple, si la carte est utilisée comme matérialisation d'un droit d'accès, la zone de transactions est superflue. Par contre, si la carte sert de dossier portable, c'est la zone de transactions qui devient la plus importante et qui sera sans doute la plus grande.

La seconde raison pour laquelle la carte est polyvalente est qu'il n'existe pas de contraintes sémantiques sur le contenu de la zone confidentielle, de la zone de transactions et de la zone de lecture. Ces zones peuvent donc recevoir n'importe quel type d'informations.

4.3. ASPECT MULTI-SERVICES

Des trois objectifs vus au chapitre trois, l'aspect multi-services est sans doute le seul dont la réalisation laisse à désirer. En effet, pour une carte multi-services, il serait souhaitable que :

- le nombre de services accessibles ne soit pas limité arbitrairement,

- chaque service puisse être totalement indépendant des autres.

Comme nous allons le voir, la carte CP8 ne possède pas tout à fait ces deux propriétés.

4.3.1. NOMBRE DE SERVICES ACCESSIBLES

Le masque décrit au chapitre trois a été conçu pour être utilisé au plus par deux émetteurs. Cela ne signifie pas pour autant que la carte ne permet la mise en oeuvre que de deux services simultanément. En effet, chaque émetteur est libre de subdiviser la mémoire de stockage qui lui a été attribuée, ce qui lui permet de proposer plusieurs services au porteur.

Le nombre de services accessibles au moyen de la carte n'est donc limité que par la taille de la mémoire disponible.

4.3.2. INDEPENDANCE DES SERVICES

Le masque étudié présente deux caractéristiques qui garantissent une certaine indépendance aux services accessibles. Ce sont, d'une part, l'existence de deux clés émetteurs différentes et, d'autre part, la présence des bits C et CA parmi les bits systèmes (cfr point 3.2.1.2. Les mots).

Toutefois, le masque étudié présente aussi certaines faiblesses :

- Toute information accessible sur présentation de la clé émetteur primaire est aussi accessible sur présentation de la clé émetteur secondaire et vice versa.

- Lorsque la carte permet l'accès à plusieurs services, une partie de la mémoire est allouée a chaque service. Malheureusement, le masque ne gère pas le partage de la mémoire. Cela est particulièrement gênant pour la zone d'accès et pour la zone de transations :

- Puisque la zone d'accès est commune à tous les services, la présentation d'une clé émetteur fausse entraîne le blocage de toute la carte alors que seul le blocage du (des) service(s) concerné(s) par la clé présentée est souhaitable.

- Les protections de la zone de transactions s'étendent à toute la zone. Il ne peut donc pas exister simultanément des informations protégées et des informations non protégées dans la zone de transactions. Cela signifie que les protections en zone de transactions doivent être identiques pour tous les services accessibles via la carte.

- A priori, les programmes d'application sont autorisés à écrire n'importe où en zone de transactions. Il n'est donc pas impossible que des données relatives à un services soient écrites, par erreur ou par malveillance, dans une partie de la zone de transactions allouée à un autre service.

CHAPITRE 5 :
LE PAIEMENT ELECTRONIQUE

5.1. LA CARTE ET LE MONDE BANCAIRE

5.1.1. Les services existants
5.1.2. Les nouveaux services
5.1.2.1. Les services à base d'informations permanentes
5.1.2.2. Les services à base d'informations évolutives
5.1.3. Les types de paiements

5.2. LE PAIEMENT DE CONTACT

5.2.1. Les acteurs et leurs responsabilités
5.2.1.1. L'émetteur
5.2.1.2. Le porteur et/ou titulaire de compte
5.2.1.3. Le vendeur
5.2.1.4. Le prestataire du service
5.2.2. La configuration du système
5.2.2.1. La carte du client
5.2.2.2. La carte applicative
5.2.2.3. Les claviers
5.2.2.4. Les écrans
5.2.2.5. L'imprimante
5.2.2.6. Le modem
5.2.2.7. L'horodateur
5.2.2.8. Le lecteur et le module de comparaison de données biométriques
5.2.2.9. L'alimentation
5.2.2.10. Les mémoires supplémentaires
5.2.3. Le scénario de l'application
5.2.4. Les options
5.2.4.1. Crédit
5.2.4.2. Adresse
5.2.4.3. Budget
5.2.5. Les traitements annexes
5.2.5.1. La consultation
5.2.5.2. L'habilitation ou la réhabilitation
5.2.5.3. Gestions diverses
5.2.5.4. La gestion des clés porteurs

V. LE PAIEMENT ELECTRONIQUE

5.1 LA CARTE ET LE MONDE BANCAIRE .

De par leurs besoins , ce sont principalement les banques qui ont stimulé le développement actuel de la carte à microprocesseur . C'est donc à travers une application bancaire que nous allons approcher les problèmes de sécurisation de tout un système où la carte trouve un rôle . Nous pourrons par la suite , juger de l'évolution des cartes pour lesquelles il est nécessaire et vital de proposer à notre société des applications non monétaires .

Grâce à sa grande capacité de mémorisation et son intelligence active , la carte à microprocesseur offre à la profession bancaire la possibilité d'améliorer les services existants et de promouvoir de nouveaux services . Nous distinguerons ensuite les types de paiement actuels .

5.1.1. LES SERVICES EXISTANTS .

Les services classiques sont : le paiement , le retrait , le virement et la consultation de comptes .

De par sa grande capacité à réagir , de façon autonome , aux évènements extérieurs : la sécurité de ces services est améliorée .

En y combinant l'accroissement des capacités mémoires : la présence dans la carte de données propres au porteur , à l'organisme bancaire et aux transactions monétaires est désormais envisageable.

Nous pouvons aussi imaginer des systèmes ou l'information est répartie . Les coûts de télécommunication sont dès lors réduits par des prises locales de décision hautement sécurisées (sans liaisons avec le central) .

Nous constaterons par la suite une plus grande souplesse d'utilisation des services classiques de la carte .

5.1.2. LES NOUVEAUX SERVICES .

Les nouveaux services concernent tout d'abord les nouvelles fonctions offertes à l'utilisateur d'une carte dont le service est classique : changement de la clé porteur , consultation de l'historique des transactions , paramétrage des crédits . Ces services utilisent ensuite la capacité mémoire disponible et la carte devient une véritable base de données . L'information qui y est stockée peut être soit permanente et/ou évolutive .

5.1.2.1 LES SERVICES A BASE D'INFORMATIONS PERMANENTES .

Les données sont alors enregistrées dans la carte , soit avant la phase active de la carte (enregistrement de l'identité bancaire et civile du porteur...) , soit pendant cette phase active (enregistrement de la tarification d'un service, d'un numéro d'affiliation , de droits d'utilisation...).

5.1.2.2 LES SERVICES A BASE D'INFORMATIONS EVOLUTIVES .

Ces données permettent de compter en nombre ou en valeur les utilisations d'un service ouvert sur la carte (prépaiement de droits unitaires , paiement ultérieur des accès réalisés sur une période...) .

5.1.3. LES TYPES DE PAIEMENTS .

A ce niveau , il est important de distinguer les notions de paiement de contact et celles de paiement à distance .

Le paiement de contact met en présence les acteurs suivants : le porteur de la carte , le vendeur de biens ou de services , prestataire de service (ex . Bancontact ,...) et la banque . Il regroupe :

- le paiement ou le prépaiement suite à la consommation de biens ou services , chez un commerçant munis d'un terminal de paiement (ou : distributeur de tickets , cabines téléphonique , ...) . Si cette transaction nécessite , l'accord du prestataire de service (via un réseau de télécommunication) : nous dirons que ce paiement se fait en-ligne , sinon elle est hors-ligne .
- le retrait d'espèces ou les virements éffectués à l'aide d'un distributeur automatique de billets .

Le paiement à distance met en présence les mêmes acteurs , mais la différence réside dans la distance qui sépare le client du vendeur . Le client possède - à domicile - un micro-ordinateur et un moyen de télécommunication ou un minitel , ce qui lui permet par exemple de disposer de la banque à domicile , d'acheter des biens à distance via un catalogue électronique , ...

5.2 LE PAIEMENT DE CONTACT .

Le but de l'application étudiée est de permettre à un client - porteur d'une carte - , de réaliser des achats chez un commerçant .

Le client possède une carte à microprocesseur , le commerçant dispose d'un système informatique pouvant entrer en dialogue avec un central . Ce système se présente sous la forme d'un terminal-caisse .

Après avoir passer en revue les acteurs et leurs responsabilités , nous étudierons la configuration du système ainsi que le scénario nécessaire au bon déroulement d'un paiement de contact .

5.2.1. LES ACTEURS ET LEURS RESPONSABILITES .

5.2.1.1 L'EMETTEUR .

Dans le cas de cette application , l'émetteur représente l'institution financière qui fournit la ligne de crédit du client titulaire du compte concerné. Il personnalise et distribue les cartes , est responsable de la mise-à-jour d'informations , du déblocage des cartes ....

5.2.1.2 LE PORTEUR ET/OU TITULAIRE DU COMPTE .

Le porteur de la carte peut en effet être une personne différente du titulaire du compte bancaire . Il doit protéger ses clés personnelles (clé principale , alternative , biométriques) qu'il a communiquées à l'émetteur lors de la personnalisation .

5.2.1.3 LE VENDEUR .

Le commerçant reçoit et accepte la carte comme moyen de paiement . Ses responsabilités concernent la protection et la mise en route du terminal-caisse ainsi que la protection des périphériques .

5.2.1.4 LE PRESTATAIRE DU SERVICE .

Il est l'organisme responsable de la fourniture des diverses cartes à microprocesseur , des différents systèmes à mettre en place chez le commerçant ou au central et enfin des programmes applicatifs réalisant le paiement de contact et les traitements annexes .

Il garantit les enregistrements des transactions comme moyen de paiement . Il agit comme intermédiaire entre les banques du client et du commerçant .

5.2.2. LA CONFIGURATION DU SYSTEME .

Le matériel de paiement réalise des fonctions nécessaires tant au déroulement de l'opération qu'au dialogue entre le commerçant , le porteur de la carte et le centre . Ce terminal-caisse est un système informatique intégré et complet disposant d'une unité centrale , d'un système d'exploitation et de mémoires . Le schéma ci-dessous représente les principaux éléments " externes " du terminal-caisse . Les éléments "internes" (unité centrale , mémoires) seront étudiées lors du chapitre 7 .

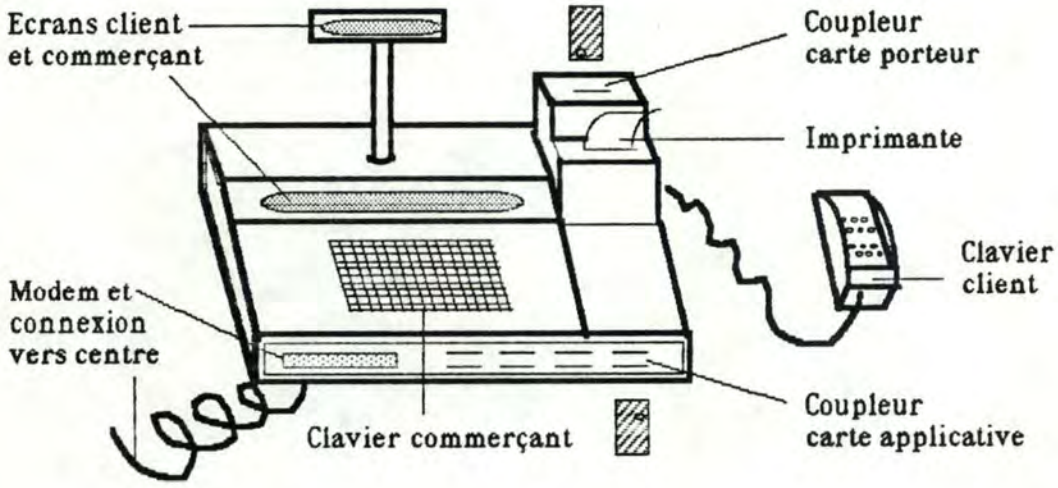


Fig. 5.1 : Le Terminal-caisse

5.2.2.1 LA CARTE DU CLIENT .

Elle est capable de vérifier l'authenticité de son porteur , du terminal caisse et de bien d'autres actions (cfr. 3.3.4. Le jeu d'instructions du microprocesseur) . Elle contient dans sa mémoire PROM :

- des données techniques relatives à la carte (sa fabrication , son contrôle d'accès ...)
- des données secrètes ou non relatives à l'émetteur des cartes (identité , clés ...)
- des données secrètes ou non relatives au porteur de la carte (identité , clés ...)
- des données relatives à l'application (validité , n°d'affiliation , plafond des achats , crédit disponible , liste des transactions réalisées jusqu'à présent)

5.2.2.2 LA CARTE APPLICATIVE .

C'est une carte spéciale , qui possède sensiblement le même jeu d'instructions qu'une carte porteur et peut jouer le même rôle qu'une carte mère (en local) (cfr. 4.1.2.2 pt. c : Confidentialité des clés , diversification).

Elle est introduite dans un connecteur de cartes . Son contenu est le suivant :

- la liste noire des clients
- le logiciel de l'application
- la table de conversion des devises
- les clés mères secrètes
- divers paramètres , tables et listes

Les informations ci-dessus sont lues et chargées dans le terminal-caisse soit en début de journée (ex. le logiciel de l'application de paiement , ...) , soit dans la journée (ex. lors de l'application) .

Le nombre de cartes applicatives varie selon celui des applications pouvant être traitées sur ce point de vente (ex. un commerçant acceptant les cartes Visa , Master Card , Eurochèque .....) .

N.B. Il est important de noter que cette caisse peut fonctionner indépendamment d'une carte à microprocesseur ou carte à bande magnétique .
Les coupleurs de la carte porteur et ceux des cartes applicatives peuvent être intégrés ou non dans le terminal-caisse .

5.2.2.3 LES CLAVIERS .

Le premier est destiné aux commerçants afin d'y taper les informations concernant une transaction (montant , ...) .

Le second est destiné aux clients afin que ceux-ci puissent introduire les données confidentielles ou non qui leur sont demandées (ex. clé porteur , type de l'achat , ...) .

5.2.2.4 LES ECRANS .

Ils sont destinés au client ou au commerçant et sont intégrés au terminal-caisse ou au clavier du client .

5.2.2.5 L'IMPRIMANTE.

Elle est utilisée pour imprimer des tickets (ou factures) lors d'une transaction.

5.2.2.6 LE MODEM.

Il est utilisé lors d'un appel au central de l'émetteur - via un réseau téléphonique - . Cet appel visant à sécuriser d'avantage une transaction . Il peut être intégré dans le terminal-caisse ou non . Les principes et les raisons de cet appel seront vus plus loin (cfr. 5.2.3.9 9° Phase) .

5.2.2.7 L'HORODATEUR.

L'horloge fournit l'heure et la date destinées au terminal .

5.2.2.8 LE LECTEUR et LE MODULE DE COMPARAISON DE DONNEES BIOMETRIQUES

Le lecteur réalise la saisie des empreintes digitales ou de la signature du client , tandis que le module s'occupe de la comparaison de ces informations avec celles présentes dans la carte porteur . Ces deux éléments sont optionnels car ils nécessitent une technologie plus avançée .

5.2.2.9 L'ALIMENTATION.

Elle fournit au terminal-caisse le voltage et la puissance requise pour tous les éléments du système . Une batterie est utilisée pour alimenter l'horloge . Si une coupure de courant survient durant une transaction , la batterie est utilisée pour terminer la transaction entamée .

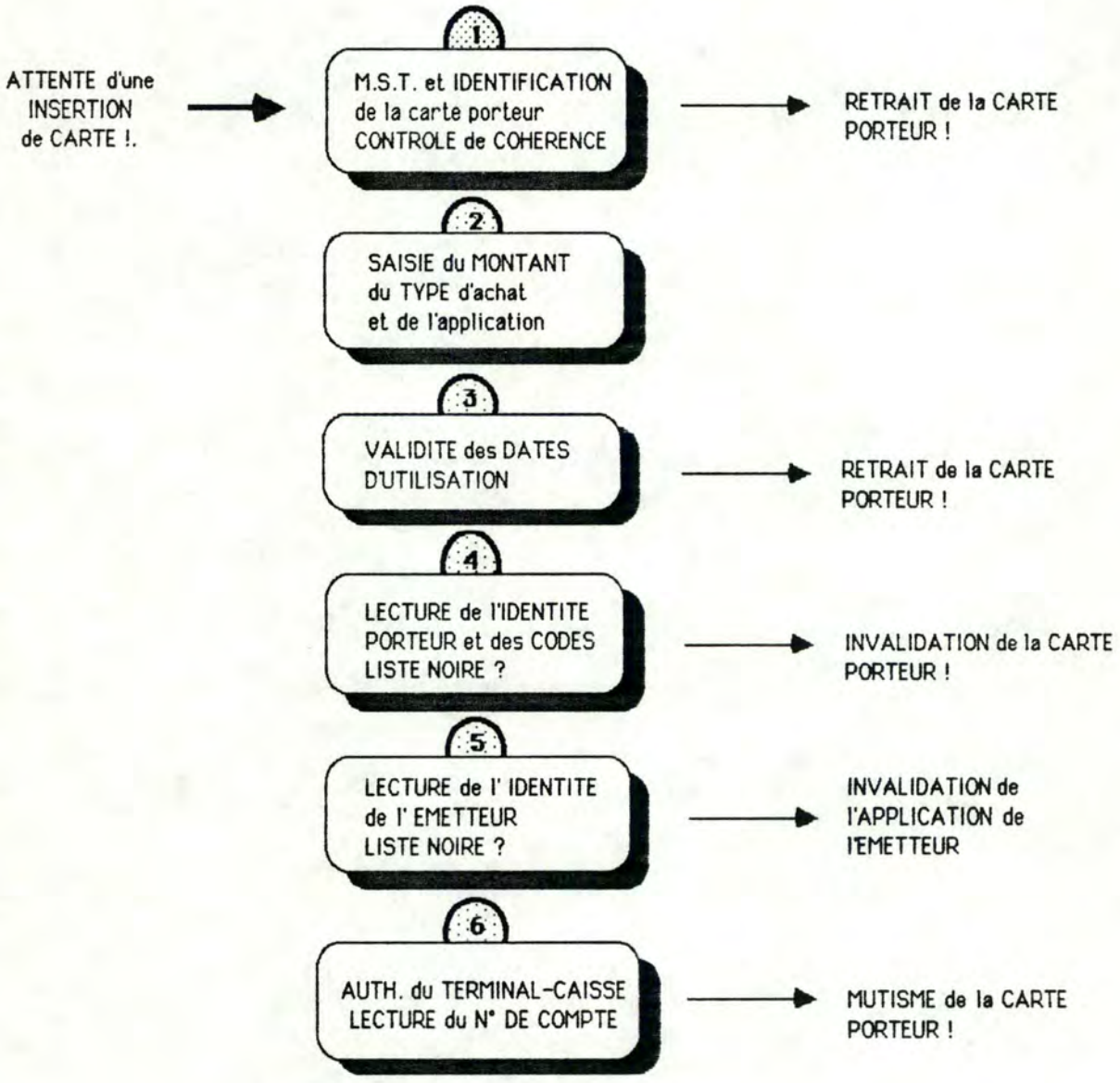
5.2.2.10 LES MEMOIRES SUPPLEMENTAIRES.

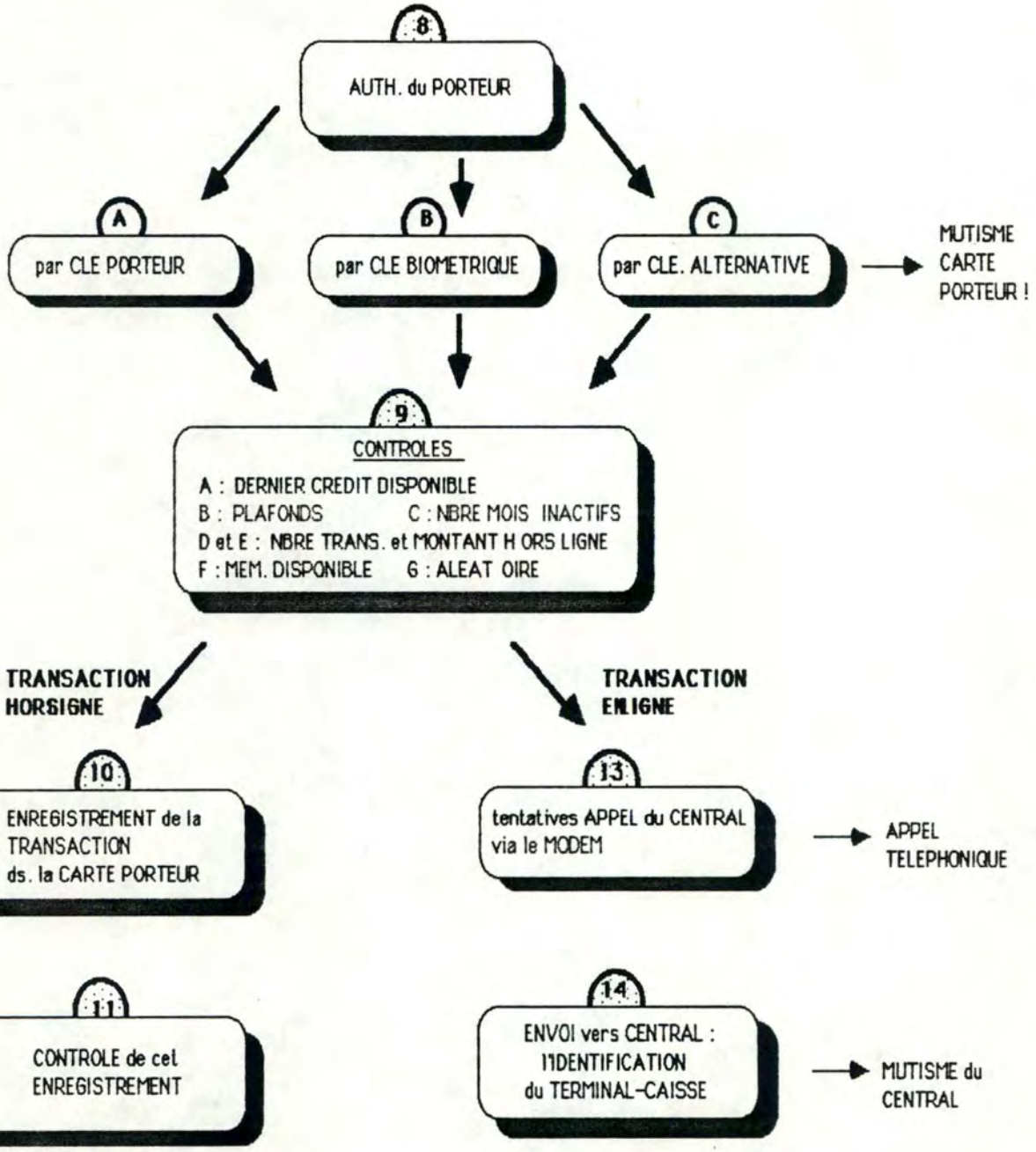
Le terminal-caisse dispose de mémoires supplémentaires . Celles-ci étant destinées entre-autre au stockage de toutes les transactions d'une journée , ces enregistrements seront alors envoyés vers le centre qui pourra ainsi débiter et créditer les comptes appropriés .

5.2.3. LE SCENARIO DE L'APPLICATION .

Le scénario suivant nous illustre à la fois la souplesse d'utilisation de la carte et jusqu'où - dans une telle application - la sécurité doit intervenir . Dès que le terminal détecte une insertion d'une carte porteur, la procédure illustrée sur les trois pages suivantes est engagée .

Les indications portées sur la droite de certaines phases et précédées d'une simple flèche représentent une rupture de la séquence normale et correspondent souvent à l'échec d'un contrôle ou d'une authentification réalisé lors de la phase courante .




8
AUTH. du PORTEUR
A
par CLE PORTEUR
B
par CLE BIOMETRIQUE
C
par CLE. ALTERNATIVE
MUTISME CARTE PORTEUR !
9
CONTROLES
A : DERNIER CREDIT DISPONIBLE
B : PLAFONDS
C : NBRE MOIS INACTIFS
D et E : NBRE TRANS. et MONTANT H ORS LIGNE
F : MEM. DISPONIBLE
G : ALEAT OIRE
TRANSACTION HORSIGNE
TRANSACTION EN LIGNE
10
ENREGISTREMENT de la TRANSACTION ds. la CARTE PORTEUR
11
CONTROLE de cet ENREGISTREMENT
13
tentatives APPEL du CENTRAL via le MODEM
APPEL TELEPHONIQUE
14
ENVOI vers CENTRAL : l'IDENTIFICATION du TERMINAL-CAISSE
MUTISME du CENTRAL

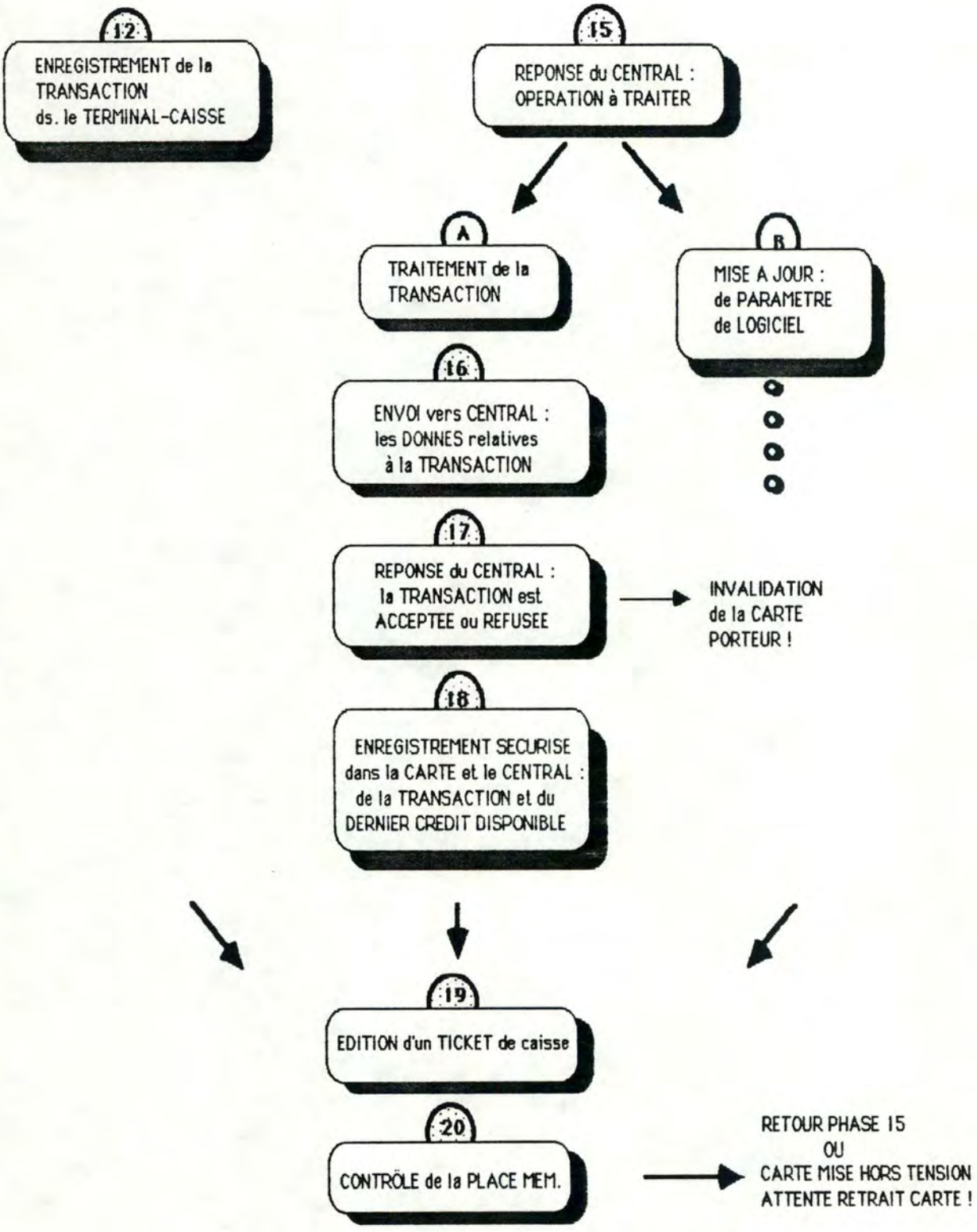


Fig. 5.2 : Le scénario du paiement de contact

5.2.3.1 1° PHASE

La première phase est l'acquisition des caractéristiques physiques et logiques du microcircuit de la carte porteur . En bref , c'est ici que :

- la carte porteur est mise sous tension afin de déterminer si elle fonctionne correctement.
- la carte porteur et le terminal déterminent leur protocole d'échange de données .
- la carte délivre son n° de série , le type d'application qu'elle peut gérer , le type de son masque .....
- la cohérence des données fournies est contrôlée , en cas d'échec , la carte doit être retirée du coupleur .

5.2.3.2 2° PHASE

Vient ensuite la saisie au clavier commerçant du montant de la transaction , de son type (détail , hôtel , aviation , ...) et de l'application choisie par le client (Visa , Master-card , ...) .

5.2.3.3 3° PHASE

Le terminal-caisse lit les dates de début et de fin de validité de la carte porteur . La date du jour , fournie par le terminal-caisse , doit être comprise entre les deux premières , sinon elle doit être retirée du coupleur .

5.2.3.4 4° PHASE

L'identité du porteur - bancaire ou non - est extraite de la carte porteur . Les différents composants de cette identité et les tests appropriés sont les suivants :

- le nom du porteur et son n° de compte : le terminal vérifie si la carte présente n'est pas frappée d'opposition (liste noire de clients) . Dans ce dernier cas : la carte est invalidée .
- le code devise : si celui de la carte porteur est différent de celui contenu dans le terminal , il y a connexion vers le central afin de se procurer le dernier taux de conversion des devises (Ex. Cas d'utilisation de la carte dans un pays étranger) .
- le code langage : de la même façon , en cas de non similitude , la connexion est nécessaire (idem) .

5.2.3.5 5° PHASE

L'identité de l'émetteur de l'application choisie par le client (Phase 2) est extraite de la carte porteur . Un contrôle des oppositions est aussi possible au niveau des émetteurs interdits de cartes . Si l'émetteur est interdit , l'application concernée doit être invalidée .

Rem .

Il est important de signaler que lors d'une certification ou d'une authentification (terminal-caisse , carte porteur , porteur) , la carte porteur doit effectuer un calcul algorithmique . En lui demandant ce calcul , avec comme paramètre : l'adresse d'une écriture ou d'une lecture antérieure ; l'application s'assure de l'écriture correcte ou de la lecture correcte d'informations dans la carte porteur.

A partir de la phase 6 , la condition du bon déroulement de la transaction est l' authentification mutuelle des trois parties suivantes : le porteur , la carte porteur et le terminal-caisse . En cas d'échec , la transaction est annulée!

5.2.3.6 6° PHASE

Authentification du terminal-caisse : tant que celui-ci ne s'est pas correctement authentifié , la carte porteur n'accepte , avec lui , qu' un dialogue partiel . Par cette authentification , la carte porteur s'assure que le terminal-caisse dans lequel elle est insérée possède bien une carte applicative appropriée et non trafiquée . Le terminal-caisse doit donc prouver la connaissance de l'algorithme Télépass et de certaines clés .

En cas de non-correspondance , la carte porteur restera muette aux ordres d'écriture , de calculs , ...

Cette authentification du terminal-caisse vise aussi à s'assurer de la réussite d'une lecture antérieure . En effet , le terminal-caisse vérifie que le n° de compte lu précédemment dans la carte porteur n'a pas été falsifié .

5.2.3.7 7° PHASE

Authentification de la carte porteur : elle est assez semblable à celle vue au point 4.1.3.3 (L'authentification de la carte) , à la différence que la carte mère est représentée ici par la carte applicative (cette authentification a lieu sans connexion avec le centre) . Cette phase consiste aussi à s'assurer de la bonne lecture du dernier crédit disponible de la carte porteur .

5.2.3.8 8° PHASE

Authentification du porteur : pour des raisons de sécurité , différentes authentifications du porteur sont prévues :

- par clé porteur
- par clés biométriques (empreintes digitales ou signature)
- par clés alternatives (questions personnelles identifiant le propriétaire) .

Ces authentifications ne sont pas exclusives , mais peuvent-être complémentaires ! Une fois cette phase terminée : le client ne peut plus annuler la transaction de son propre gré

A: L'authentification du porteur par clé porteur .

Cette authentification est similaire à celle déjà vue au point 4.1.3.1 (Authentification du porteur) . Elle consiste en une simple présentation de la clé saisie au clavier du client . Il peut , néanmoins , être possible de fournir cette clé saisie de façon chiffrée . La zone d'accès de la carte porteur permet de limiter le nombre de présentations de clés porteur érronées (trois essais érronés consécutifs invalidant la carte porteur) .

B: L'authentification du porteur par clés biométriques .

Cette authentification consiste en la comparaison des empreintes (ou de la signature) présentées par l'utilisateur de la carte porteur avec les empreintes (ou signature) présente dans cette carte .
Ces informations sont digitalisées , d'ou la nécessité d'appareils sophistiqués.

C: L'authentification du porteur par clés alternatives .

La carte est capable de stocker dans sa mémoire PROM , une série de réponses identifiant le porteur de la carte . La(les) question(s) est(sont) soumise(s) au porteur et controlée(s) de la même façon que pour la clé porteur .

5.2.3.9 9° PHASE

Les contrôles suivants servent à déterminer si la transaction commençée doit se poursuivre en mode local (hors-ligne) ou en connexion avec le centre (en-ligne) . Ils visent soit à limiter le pouvoir d'achat hors ligne des clients , soit à les prévenir d'évènements , soit à se prémunir des fraudeurs .

Les situations ou le contrôle échoue sont des cas litigieux , auquel cas : il est nécessaire pour le terminal-caisse de laisser la prise de décision au central . L'échec d'un contrôle n'entraîne pas automatiquement le refus de la transaction courante .

Rem .

Les divers plafonds cités sont stockés dans la carte porteur ! Tous ces contrôles sont réalisés le terminal-caisse . D'autres contrôles peuvent être aisément implémentés (ex. contrôle du cumul des derniers jours .

A: Le contrôle du dernier crédit .

Ce premier contrôle consiste à vérifier que le montant de la transaction courante n'excède pas le dernier crédit du client diminué de la somme des achats qu'il a effectués après la date de ce crédit .

B: Le contrôle des plafonds .

Suivant le type de transaction introduit lors de la phase 2 : un plafond déterminé ne doit pas être dépassé par le montant de la transaction courante . Ces différents plafonds sont présents dans la carte porteur ; mais de la même façon , le montant peut être comparé à un des plafonds présents dans la carte applicative . Cette démarche vise à limiter la capacité d'achat des clients .

On peut aussi laisser , au commerçant , la possibilité de passer outre ses plafonds ; dès lors , il ne bénéficie plus des garanties bancaires .

C: Le contrôle du nombre de mois d'inactivité .

La carte porteur ne peut pas rester inutilisée plus d'un certain nombre de mois .

D: Le contrôle du dernier nombre de transactions réalisées hors ligne .

Si la dernière transaction est hors ligne , il faut vérifier que la limite autorisée de transactions consécutivement hors ligne n'est pas atteinte .

E: Le contrôle du montant de ces dernières transactions hors ligne.

Le montant total des dernières transactions consécutivement hors ligne ne doit pas dépasser un certain plafond .

F: Le contrôle de la mémoire disponible .

Une place suffisament grande , dans la zone des transactions , doit être disponible afin d'y stocker la transaction courante .

G: Le contrôle aléatoire .

Il est aussi intéressant de prévoir une connexion aléatoire au centre afin de décourager les fraudeurs plus avertis .

Rem .

La phase suivante va dépendre de la réussite de ces divers contrôles . En effet , si un seul de ces contrôles a échoué , la transaction doit être continuée en présence du central (transaction en-ligne) . Dans le cas contraire , la transaction est acceptée et se termine hors-ligne .

1° CAS : LA TRANSACTION RESTE HORS LIGNE .

5.2.3.10 10° PHASE

La transaction est acceptée car tous les contrôles ont échoué , elle est enregistrée dans une zone mémoire de la carte porteur (la zone des transactions) . Cet enregistrement est principalement composé des champs suivants :

- la date du jour
- le montant
- le nom du commerçant ....

5.2.3.11 11° PHASE

Cette phase consiste pour le terminal-caisse , à s'assurer que le montant de la transaction a correctement été enregistré dans la carte porteur (lors de la phase 10) .

Le scénario est semblable à celui vu au point 4.1.3.3 (La certification) . La carte porteur doit délivrer au terminal-caisse un certificat calculé à partir d'un nombre aléatoire et de l' adresse du montant à certifier . Le terminal-caisse , grâce à sa carte applicative est capable de vérifier si ce certificat est correct .

5.2.3.12 12° PHASE

Comme nous l'avons expliqué précédemment , chaque transaction est stockée dans une mémoire du teminal-caisse , afin d'être transmise ultérieurement vers le central . Cet enregistrement comprendt :

- un n° d'ordre
- le n° de série carte
- la date et l' heure de la transaction
- le montant
- le n° de compte du client
- le certificat (cfr. Phase 11)
- le nombre aléatoire et adresse utilisés pour ce certificat

Le commerçant , est assuré par la détention du certificat que le client ne niera pas avoir présenté sa propre carte porteur .

De même : le client , grâce à ce certificat , est assuré qu'un faux terminal-caisse ne pourra plus modifier le montant de la transaction ou l'identifiant du client (n° de compte) . En effet , seul un vrai terminal-caisse est capable - grâce à la carte applicative - de calculer un certificat correct .

2° CAS : LA TRANSACTION EST EN LIGNE .

5.2.3.13 13° PHASE

La première opération consiste à établir la connexion avec le centre - via le modem - . La carte applicative concernée possède pour cela un numéro d'appel téléphonique qu'elle communique au modem .

En cas d'échec après x tentatives de connexions , le commerçant doit établir une connexion orale avec le centre via le réseau téléphonique . Le terminal-caisse lui indique alors les opérations à suivre ainsi que les informations à communiquer oralement . Ce type de communication ne sera pas abordé dans ce document .

5.2.3.14 14° PHASE

Une fois la connexion - via modem - établie : le terminal-caisse est prié de s'identifier auprès du centre . Il utilise le même procédé que lors de la phase 6 (Authentification du terminal-caisse) . En cas d'échec , le central reste muet .

5.2.3.15 15° PHASE

Le central répond à l'appel du terminal-caisse en déterminant la première opération à réaliser . Avant le traitement de la transaction courante ; le central peut demander au terminal-caisse , l'exécution d'une des deux opérations suivantes :

- mise à jour de paramètres contenu dans la carte applicative
- chargement d'un logiciel applicatif dans la carte applicative

Une fois la première opération réalisée , le central décide à nouveau de l'opération suivante . Tant que le central désire réaliser une opération , une itération (Phase 15 à 20) est exécutée .

La mise à jour de paramètres ou de logiciels ne sera pas traitée ici !

5.2.3.16 16° PHASE

Dès que le central , a accepté de traiter la transaction courante : le terminal réunit les informations suivantes afin de les lui communiquer :

- les n° de série de la carte porteur et applicative concernée
- le numéro de compte du client
- le montant de la transaction
- la raison pour laquelle , la connexion en ligne a été provoquée (identifiant du contrôle qui a échoué)
- l'estimation des derniers montants hors ligne réalisés avec la carte porteur

5.2.3.17 17° PHASE

C'est , sur base des informations fournies ci-dessus et d'autres que le central décide ou non d'accorder la transaction courante .

La réponse du terminal comprend :

- l'accord ou le refus de la transaction identifié , par un numéro de contrôle . En cas de refus , la raison de celui-ci (invalidation possible de la carte)
- le nouveau crédit disponible chiffré avec une clé

5.2.3.18 18° PHASE

Dans le cas d'un accord , il est procédé à l'écriture de la transaction dans la carte porteur de la même manière que celle vue à la phase 10 et 11 .

Le central en profite aussi pour mettre à jour le dernier crédit disponible de la carte porteur . Cette mise à jour est sécurisée de la même façon qu'une signature électronique (cfr. point 4.1.3.4) . La carte porteur n'accepte d'écrire cette information dans sa mémoire que si la version chiffrée est en cohérence avec la version non chiffrée .

L'enregistrement destiné au terminal-caisse (cfr. phase 16) est inutile car la transaction a lieu en connexion directe avec le central .

Les dernières phases communes aux transactions en-ligne et hors-ligne sont les suivantes :

5.2.3.19 19° PHASE

Edition d'un ticket de caisse : celui-ci peut aussi servir comme justificatif de paiement grâce aux informations qu'il contient :

- nom , raison social du commerçant
- date et heure de la transaction
- montant de la transaction
- numéro de série de la carte porteur
- numéro d'identification de l'accord du centre
  (en cas de connexion en-ligne , cfr. réponse phase 17)

5.2.3.20 20° PHASE

En cas de saturation attendue de la mémoire de la carte porteur (place encore disponible pour moins de 10 transactions) , le terminal-caisse invite le client à se rendre dans sa banque afin de se procurer une nouvelle carte . Si le central ne désire plus utiliser la carte , elle est mise hors tension , puis retirée.

Rem .

Au cas ou le contrôle E de la 9° phase échoue , la transaction peut-être acceptée par le central sans qu'aucune informations ne soient écrites dans la carte porteur .

5.2.4. LES OPTIONS .

Il peut être intéressant pour le client de disposer de services supplémentaires , dans l'application de paiement de contact . Ce sont les options crédit , adresse et budget que le client décide d'acquérir avant la personnalisation de sa carte .

5.2.4.1 CREDIT .

Soit un client bénéficiant de l'option crédit . Si lors d'une transaction , il émet le désir d'en profiter : le terminal-caisse procède au calcul du pouvoir d'achat (à crédit) résiduel ainsi que le pouvoir d'achat (au comptant) résiduel .

Il les communique alors au client ; qui décide quelle part de transaction il veut éffectuer à crédit .

5.2.4.2 ADRESSE .

Il peut parfois être intéressant pour le commerçant de connaître l'adresse de ses clients . Si cette adresse figure en mémoire carte et que le client donne l'accord à cette divulgation , elle est communiquée au commerçant .

5.2.4.3 BUDGET .

Le client peut introduire - lors de chaque transaction - un code indiquant le type de frais associé à l'achat qu'il effectue (ex . 1 pour ménage , 2 pour voiture , ...) . Ce code , appelé code budget lui permettant de visionner ultérieurement la liste de ses achats triée sur ce critère (cfr. 5.2.5.1 La consultation) .

5.2.5. LES TRAITEMENTS ANNEXES .

La liste de certains traitements , satellites de l'application de paiement de contact , est énoncée ci-dessous . Ces traitements annexes sont ausssi des applications utilisant les cartes et sont nécessaires à l'application bancaire .

5.2.5.1 LA CONSULTATION .

Cette opération consiste à prendre connaissance des informations écrites dans la carte porteur . Les zones secrètes ne peuvent être en aucun cas consultées . Les zones accessibles de l'extérieur , mais protégées ; peuvent être consultées selon les règles définies lors de la personnalisation .

Le porteur , se présentant devant un terminal approprié , peut alors visionner ou imprimer :

- le dernier crédit disponible inscrit dans sa carte
- la liste de toutes les transactions de la carte
- la liste des transactions triée selon des critères

L'application de consultation utilise les instructions de base de la carte pour accéder à ses informations .

5.2.5.2 L'HABILITATION OU LA REHABILITATION .

Nous avons vu que : suite à la présentation consécutive de trois clés porteurs fausses , la carte est rendue inutilisable pour la plupart des instructions cartes .

Grâce à un terminal spécifique : il est possible de réhabiliter une carte dont la zone d'accès n'est pas totalement consommée . Pour cela , il faut présenter successivement la clé porteur valide et la clé émetteur valide .

Dans le cas d'une saturation trop rapide de la zone d'accès , on peut étendre cette dernière .

5.2.5.3 GESTIONS DIVERSES .

Les divers plafonds ainsi que les autres seuils d'appel au centre (cfr. Phase 9) sont introduits lors de la personnalisation de la carte ! Il est néanmoins utile de pouvoir les modifier durant la phase active de la carte . C'est lors d'un transaction en-ligne que ces divers paramètres peuvent être mis à jour , cette écriture est réalisée par le central grâce à une clé appropriée .

5.2.5.4 LA GESTION DES CLES PORTEURS .

Le porteur doit avoir la possibilité de changer sa clé porteur à un nombre défini de reprise .

Le scénario est le suivant :

- le porteur après avoir inséré sa carte , s'identifie en présentant sa clé porteur , ses empreintes ou son identificateur alternatif . Dès lors : si le porteur présente trois fois consécutivement et correctement sa nouvelle clé porteur, elle est acceptée et envoyée vers le central qui est le seul capable de l'écrire - à distance - dans la zone secrète de la carte . Il utilise pour cela le chiffrement et sa clé émetteur . L'ancienne clé porteur devient alors inutile .

CHAPITRE 6 :
L'EVOLUTION DES CARTES

6.1. LA DESCRIPTION DU MASQUE M.A.

6.1.1. L'organisation de la mémoire PROM.
- 6.1.1.1. Les mots mémoires
- 6.1.1.2. Les zones
  - A. Zone de fabrication
  - B. Zone secrète
  - C. Zone d'accès
  - D. zone de travail
  - E. Zone publique
  - F. L'accessibilité des zones
- 6.1.1.3. Les niveaux
- 6.1.1.4. La gestion des niveaux et des zones
  - A. Le dynamisme des niveaux et des zones
  - B. Le principe d'exploitation des niveaux et des zones

6.1.2. Le jeu d'instructions du microprocesseur
- 6.1.2.1. Instructions d'initialisation
- 6.1.2.2. Instructions simples
- 6.1.2.3. Instructions basées sur l'algorithme Télépass

6.2. LES AVANTAGES DU MASQUE M.A.

6.2.1. Les avantages apportés par le jeu d'instructions
- 6.2.1.1. Les instructions simples
- 6.2.1.2. Les instructions algorithmiques
  - A. Auth. du porteur
  - B. Auth. du terminal
  - C. Auth. de l'émetteur
  - D. La certification et l'auth. de la carte
  - E. La téléécriture

6.2.2. Les avantages apportés par la gestion mémoire
- 6.2.2.1. La polyvalence
- 6.2.2.2. Multi-services

6.3. CONCLUSIONS

VI. L'EVOLUTION DES CARTES.

Au cours de ce chapitre, nous proposons de mettre à jour les insuffisances et faiblesses du masque de la carte étudié au point 3.2. (Description d'un masque). Le rôle de l'application bancaire est de nous faire comprendre l'accroissement des exigences envers la carte à microprocesseur. Exigences se situant à différents niveaux : celui de la sécurité , celui de la souplesse d'utilisation ou encore celui de l'organisation des mémoires .

Le nouveau masque noté masque MA et décrit ci-dessous , nous renseigne sur l'évolution actuelle des cartes à microprocesseur ! Nous verrons ensuite en quoi ce masque comporte des avantages .

6.1. LA DESCRIPTION DU MASQUE M.A. .

Ce masque illustrent l'évolution actuelle des cartes . Il sera surtout étudié dans le contexte d'une phase active . La carte est donc supposée être assemblée et personnalisée . La fabrication et dans une moindre mesure la personnalisation , se basent sur les mêmes principes que ceux vus au point 3.4 (Le cycle de vie de la carte) et au point 4.1.2.2.c (La confidentialité des clés : la diversification) .

L'étude de ce masque abordera deux aspects:

- l'organisation de la mémoire PROM (gestion et accès des mémoires)
- Son jeu d'instructions (les fonctionnalités de la carte)

6.1.1. L'ORGANISATION DE LA MEMOIRE PROM .

Cette organisation vise le partage de la mémoire en une suite d'espaces de relativement autonomie . Les deux caractéristiques énumérées ci-dessous y contribuent .

La première caractéristique est la hiérarchisation des mémoires en trois niveaux : les niveaux carte , application et service . Ainsi , une carte peut abriter une ou plusieures applications et une application peut abriter un ou plusieurs services .

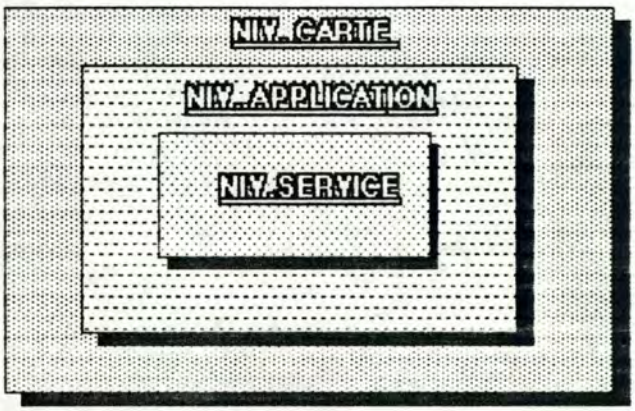


Fig.6.1 : Les niveaux de la mémoire.

L'autre caractéristique plus connue est la notion de zones : à chaque niveau, nous pouvons rencontrer les cinq types de zones suivantes : zone de fabrication (Z.F.) , zone sécurité (Z.S.) , zone d'accès (Z.A.) , zone publique (Z.P.) ou zone de travail (Z.T.) .

Malgré certaines contraintes étudiées plus loin : les combinaisons niveaux-zones sont très souples . La figure 7.2. nous illustre des configurations possibles de trois cartes dont la mémoire PROM. est gérée par ce nouveau masque :

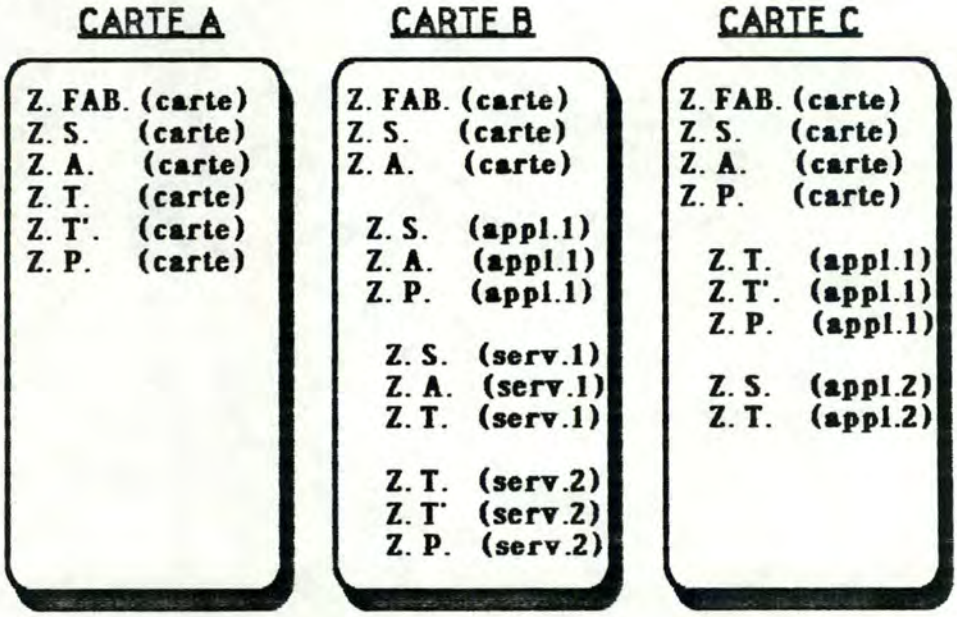


Fig.6.2 : Des combinaisons niveaux-zones.

Mis à part la zone de fabrication (Z. F.) , qui est présente uniquement et obligatoirement au niveau carte , les autres zones sont combinables à tous niveaux .

Cette organisation des mémoires nécessite diverses innovations et modifications des concepts fondamentaux vus au point 3.2.1 (Gestion de la mémoire de stockage PROM.) , soit :

- les mots mémoires
- les zones
- les niveaux

Après leurs études , nous concluerons la description de cette nouvelle mémoire PROM par la gestion des niveaux et des zones .

6.1.1.1 LES MOTS MEMOIRES

Un mot mémoire comporte toujours 32 bits ; mais , comme l'illustre la figure 6.3. son contenu varie.

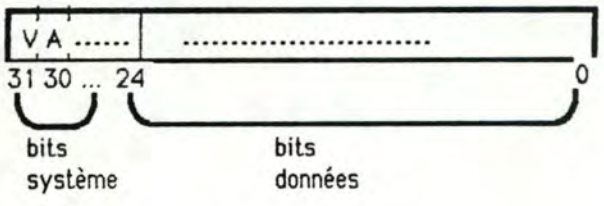


Fig. 6.3 : Le contenu d'un mot mémoire.

Le nombre (min.3 , max.8) et la structure des bits système (bits de poids forts) varient selon le contenu des bits de données (bits de poids faibles) .

A. Les bits système .

Le tableau ci-dessus illustre leurs significations.

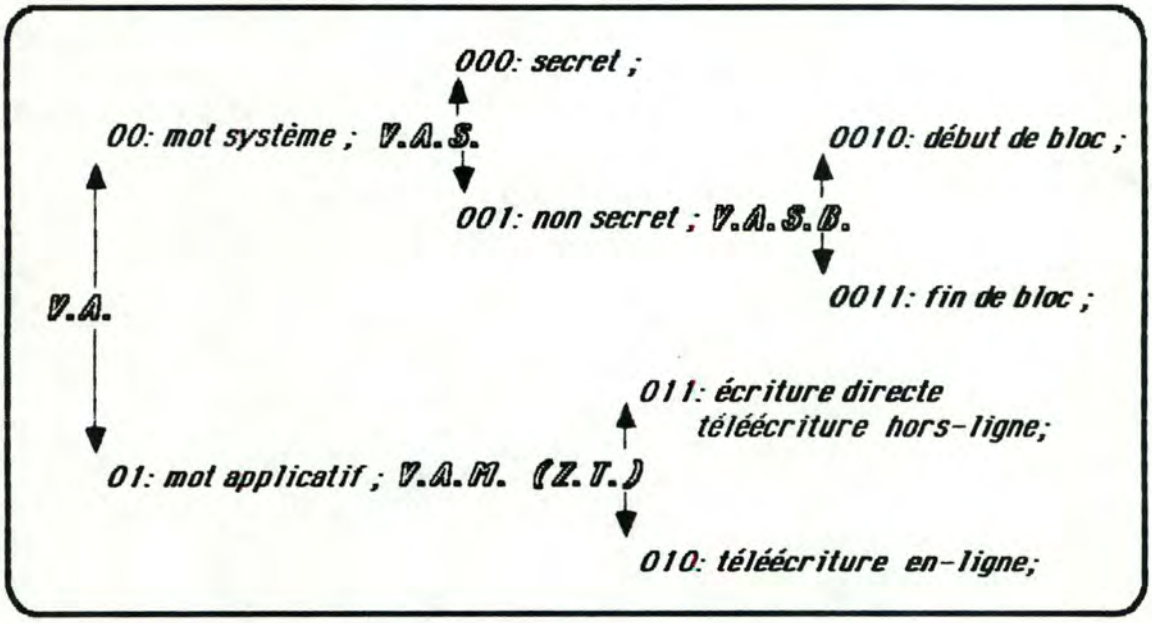


Fig. 6.4 : Les bits système.

- Le bit V indique si le mot est validé (o) ou non (1) .
- Le bit A indique si le mot est de type système (0) ou applicatif (1) .
- Le bit S indique si le mot a été écrit en-ligne (0) ou non (1) .
- Le bit B indique si le mot est en début de bloc descriptif (0) ou non (1) .
- Le bit M indique si le mot est secret (0) ou non (1) .

B. Les bits données.

Mis à part dans les zones d'accès , les bits de données peuvent abriter les informations suivantes :

1) si VAS = 000 , les 29 bits de données abritent une clé .
2) si VAM = 0IX , Les 29 bits de données de ce mot (contenus dans la zone de travail) sont dépendants de l'application .
3) si VASBNIZO = 00I0XXXX , le mot est dit "descriptif" et les 4 bits qui suivent VASB indiquent alors le niveau (bits NI) et la zone (bits ZO) où se trouve ce mot . Les 24 bits de données sont utilisés pour décrire la zone (ZO) qui suit .
4) si VASB = 00II , les 28 bits de données sont utilisés dans le même but que ci-dessus . Ce mot est utilisé avec le mot précédent lorsqu'il est insuffisant pour décrire la zone .

6.1.1.2 LES ZONES .

Les rôles et le contenu des diverses zones (zone fabrication , secrète , d'accès , publique et de travail) , bien qu'étant presque semblables à ceux-vu lors du premier masque sont explicités ci-dessous .

A. Zone de fabrication.

Elle contient sensiblement les mêmes informations sur l'identité technique de la carte . Elle ne possède plus de données concernant la taille et la localisation des autres zones . La clé de fabrication est désormais stockée dans une zone secrète .

B. Zone secrète.

Le rôle de cette zone est toujours d'abriter les diverses clés secrètes . Voyons successivement la liste des clés , leur multiplicité , leur localisation dans les niveaux , la notion de version d'une clef , l'enregistrement de celle-ci et enfin la notion d'entête d'une zone secrète .

Description des clés.

- La **clé émetteur** , l'émetteur est responsable d'un (de) niveau(x) (son application avec son (ses) service(s)) . Il peut exister plus de deux émetteurs présents sur une carte : ceux-ci disposent alors d'une clé distincte . Elle leur permet de fixer les droits et limites d'utilisation du niveau dans lequel la clé se trouve . Elle les autorise aussi à débloquer le niveau , à créer des zones dans ce niveau et des niveaux inférieurs au niveau courant .

- La **clé de certificat** est utilisée pour le simple calcul de certificat et pour l'authentification de la carte .

- La **clé système** joue un rôle dans l'authentification du terminal, du porteur et le chiffrement d'informations .

- La **clé porteur** indentifie le porteur qui peut ainsi utiliser les applications et services mis à sa disposition .

- La **clé alternative** , ainsi que la **clé biométrique** sont des moyens de sécuriser de façon accrue la phase de reconnaissance du porteur . La clé alternative est un complément à la clé porteur et est généralement biographique (ex. - date de naissance fournie par le porteur de la carte) . De même , le porteur peut-être amené à fournir ses empreintes digitales ou sa signature (informations biométriques) , on utilise alors pour son authentification une clé biométrique .

Multiplicité et localisation .

Vu le nombre d'applications et de services disponibles dans une carte , nous y retrouvons de mutiples clés d'un ou l'autre type :

Ex: - une clé émetteur par application .
- une clé certificat pour l'application n°1 et une clé certificat pour le premier service de cette application .
- une application sans clé système utilisant celle du niveau supérieur .

Toutes ces clés sont situées dans la zone secrète du niveau concerné (carte , application , service) . Un niveau ne possèdant pas un type de clé se réfère au niveau supérieur lorsque cette clé est concernée .

Les clés relatives au porteur (clé porteur , clé alternative , clé biométrique) de la carte sont localisées dans la zone secrète du niveau carte . Les autres clés (émetteur, de certificat ou de système) peuvent être attachées à une application ou à un service particulier .

Version de clé.

Chaque niveau carte , application ou service peut posséder différentes versions d' un type de clés ; mais elles ne sont pas toujours toutes actives au même instant .

Ex : - deux clés porteur dans la carte mais une seule est active .
- une application peut posséder diverses clés système actives simultanément .

Enregistrement des clés.

Dans un premier temps , on personnalise la carte de façon globale ; puis chaque émetteur personnalise par la suite sa propre application ou service .

Les clés relatives au porteur sont introduites lors de la personnalisation globale de la carte . Les autres clés (émetteur , de certificat , système) sont enregistrées lors de la personnalisation du niveau abritant l'application ou le service .

Entête d'une zone secrète.

Nous avons vu ci-dessus, que les zones secrètes pouvaient se retrouver à différents niveaux et contenir différentes clés .

Pour cette raison et dans le but d'une meilleure gestion de la mémoire : la zone secrète est représentée par une suite de paires d'éléments . Chaque paire constituée de la clé secrète et de son entête .

La figure suivante illustre un exemple de zone secrète pouvant se situer au niveau d'une application .

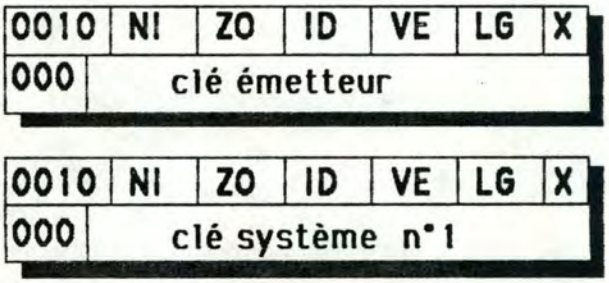


Fig. 6.5 : Entête d'une zone secrète .

L'entête permet d'identifier la clé et sa localisation de la façon suivante :

NI: le niveau ou la zone secrète se situe
ZO: le type de zone identifiée par cette entête (zone secrète) .
ID: l'identification de la clé qui suit l'entête (porteur, émetteur)
VE: le n° de version de cette clé .
LG: la longueur de cette clé .

C. Zone d'accès.

Cette zone est toujours utilisée par le microprocesseur , afin d'y mémoriser toutes tentatives d'accès correctes ou non à une zone protégée .Les accès relatifs aux clés certificat ne sont pas comptabilisés dans une zone d'accès .

La zone d'accès gérant les clés porteurs se situe obligatoirement au niveau carte . Vu la multiplité des autres clés d'accès (émetteur , système) , nous pouvons trouver à tous niveaux des zones d'accès .

De même que pour la détention des clés : si un niveau ne possède pas de système de contrôle des accès , on se réfère au niveau supérieur . L'entête d'une zone d'accès est illustrée ci-dessous :

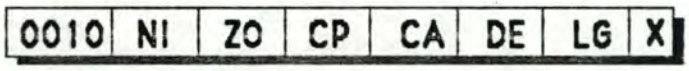


Fig. 6.6. Entête d'une zone d'accès .

NI: le niveau où la zone d'accès se situe .
ZO: le type de zone identifiée par cette entête (zone d'accès) .
CP: le nombre d'essais par clé porteur autorisé avant blocage .
CA: Le nombre d'essais , par les autres clé d'accès , autorisé avant blocage .
C.D: les conditions de déblocage de cette zone d'accès .
L.G: la longueur de la zone d'accès .

D. Zone de travail.

Elle contient les informations enregistrées lors des opérations cartes de l'application (ex: Lors d'une application bancaire : les transactions monétaires et les derniers crédits disponibles) .

Une application (ou un service) peut toujours posséder plus d'une zone de travail .

Cette zone est précédée de deux mots d'entête , leur contenu est le suivant:

0010	NI	ZO	ID	LG1	X
0011	CE	PE	PL	LG2	X

Fig. 6.7 : Entête d'une zone de travail.

NI: le niveau où la zone de travail se situe .
ZO: le type de zone identifié par cette entête (zone de travail).
ID: l'identifiant cette zone de travail dans le niveau .
LG1-LG2: la longueur de la zone de travail .
CE: les conditions d'effacement de la zone de travail .
PE: les protections en écriture de la zone de travail .
PL: les protections en lecture de la zone de travail .

E. Zone publique.

Cette zone abrite des informations non confidentielles en provenance de l'émetteur. (ex: n° de compte en banque , adresse...) . Son entête est la suivante :

0010	NI	ZO	ID	LG	X

Fig. 6.8 : Entête d'une zone publique.

NI: le niveau où la zone publique se situe .
ZO: le type de zone identifiée par cette entête (zone publique) .
ID: l' identifiant de cette zone publique dans le niveau .
LG: la longueur de la zone publique .

Le contenu des cinq zones est résumé à la figure 6.9.

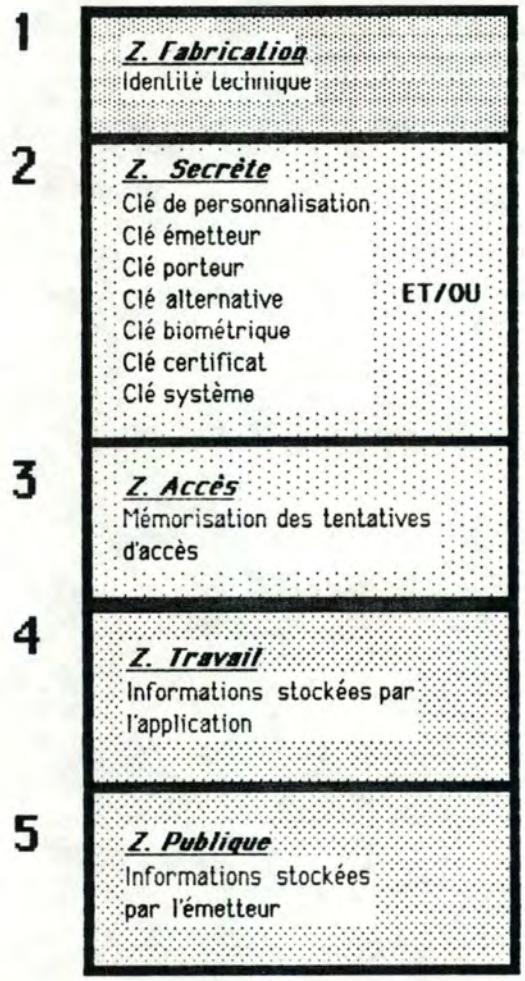


Fig. 6.9 : Le contenu des cinq zones.

F. L'accessibilité des zones.

Pour le monde extérieur (les programmes d'application) , la lecture et l'écriture des cinq zones sont soumises aux règles suivantes :

	ACCESSIBILITE	
	EN LECTURE	EN ECRITURE
ZONE FABRICATION	oui	non
ZONE SECRETE	non	oui 1
ZONE D'ACCES	oui	non
ZONE TRAVAIL	oui 2	oui 3
ZONE PUBLIQUE	oui	oui 4

Fig. 6.10 : L'accessibilité des zones.

Oui: la lecture ou l'écriture sont totalement libres .
1: seule la téléécriture est autorisée .
2: la lecture dépend des conditions PL , indiquées dans l'entête de la zone de travail .
3: l'écriture des conditions PE .
4: l'écriture est autorisée après présentation de la clé émetteur.

Le microprocesseur possède deux prérogatives supplémentaires à un programme d'application :

- Ecriture de la zone secrète .
- Ecriture dans la zone d'accès .

6.1.1.3 LES NIVEAUX.

La mémoire peut être partagée entre différents organismes (ORG.) et ces derniers ont la possibilité d'établir entre eux une certaine hiérarchie .

Cette hiérarchie des organismes établit grâce aux niveaux (carte , application , service) est illustrée à la figure ci-dessous .

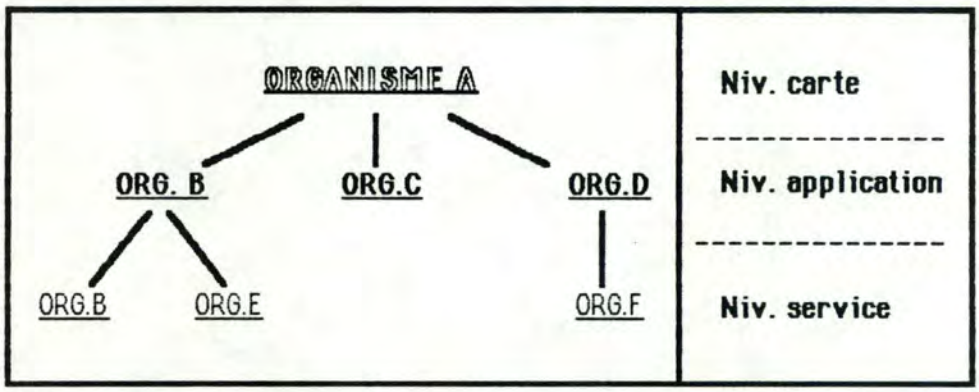


Fig. 6.11 : les niveaux .

S'il le désire , un organisme possédant - en suffisance - un espace mémoire peut en léguer une partie à un autre organisme . Le premier organisme cité détermine l'autonomie attribuée à l'organisme qui lui est inférieur . Cette autonomie est enregistrée au début du niveau abritant l'organisme grâce à une entête d'allocation et de contrôle . Le schéma suivant illustre son contenu:

0010	NI	ZO	ID	LG1	NC	T	X
0011	X	CNC	CTC	LG2	S	I	X

Fig. 6.12 : Entête d'une zone d'allocation et de contrôle.

NI: le niveau où l'entête se situe .
ZO: le type de zone identifiée pour cette entête (zone d'allocation et de contrôle) .
ID: l' identifiant de l'organisme possesseur de ce niveau .
LG1-LG2: la longeur du niveau alloué à cet organisme .
NC: le type de niveaux inférieurs pouvant être créé à partir du niveau courant (NI).
CNC: les conditions de création des niveaux inférieurs (cfr. N.C.) .
CTC: les conditions de création d'une nouvelle zone de travail à ce niveau (NI).
S: bit d'état du niveau : soit actif ou soit en personnalisation .
T: la reconnaissance du terminal est-elle exigée pour ce niveau .
I: le niveau est-il invalide ou non ?

6.1.1.4 LA GESTION DES NIVEAUX ET DES ZONES

Après un description assez technique des niveaux et des zones , il est important de s'attarder sur leur gestion et leur principe d'exploitation .

A. Le dynamisme des niveaux et des zones.

Bien qu'assez proches , 2 aspects de cette gestion sont distinguables :

- la création de zones (et de niveaux) .
- la mise à jour de niveaux .

La création de niveaux et de zones.

La création de zones peut être effectuée dans 2 contextes :

Le premier context est celui d'un niveau actif désirant se créer de nouvelles zones . Cette création consiste tout simplement en l'écriture de l'entête de la zone concernée : zone secrète , d'accès , publique , de travail (cfr. CTC)

Le second est celui d'un niveau actif (carte ou application) désirant créer et personnaliser un niveau qui lui sera inférieur (application ou service) . Cette opération se décompose en plusieures étapes :

- Création de niveaux inférieurs : l'application écrit l'entête d'allocation et de contrôle de niveau inférieur .
- Création des zones de ce niveau inférieur : les entêtes des diverses zones sont écrites de la même façon que ci-dessus .
- Enregistrement du contenu de certaines zones : cette opération est réalisée grâce à l'écriture directe de mots applicatifs ou la téléécriture de mots secrets (clé émetteur, système, ou certificat) .
- Poitionnement du bit S dans l'entête d'allocation et de contrôle du niveau: cette opération clôture la personnalisation du niveau et lui confère sa propre autonomie .

La mise à jour de niveau .

Durant la phase active d'un niveau , celui-ci est autorisé à étendre ou réduire son propre espace .

Dans le cas où un niveau est saturé , un mot de chaînage (à la suite de l'entête de ce niveau) permet au niveau d'agrandir son espace .

Dans le cas inverse , un mot de réduction (à la suite du mot de chaînage) autorise ce niveau à libérer une partie de son espace mémoire afin de le restituer au niveau supérieur.

B. Le principe d'exploitation des niveaux et des zones.

Afin de pouvoir travailler dans une zone quelconque d'un niveau . Il est d'abord , nécessaire de sélectionner cette zone . De plus , si la situation hiérarchique de la zone l'exige ; il est au préalable nécessaire de sélectionner tous les niveaux supérieurs à celui où est localisée la zone à atteindre .

Cette opération est plus ou moins semblable à l'ouverture d'un fichier . La mémoire pouvant abriter une structure de "fichiers en arbre" ; l'accès à ceux-ci doit respecter cette structure hiérarchique .

6.1.2. LE JEU D INSTRUCTIONS DU MICROPROCESSEUR .

Le jeu d'instructions de ce masque , tout en étant plus complet que celui vu lors du chapitre 3 conserve les mêmes catégories .

- instructions d'initialisation
- instructions basées sur l'algorithme Télépass
- instructions simples .

6.1.2.1. INSTRUCTIONS D'INITIALISATION.

A. Mise sous tension (**M.S.T.**) :

Son rôle reste l'initialisation du dialogue avec la carte, mais les paramètres retournés par cette instruction sont plus nombreux et plus précis .

6.1.2.2. INSTRUCTIONS SIMPLES.

A. Recherche sur arguments (**RECH.**) :

L'utilisation de cette instruction est la recherche d'un mot dans la mémoire de la carte ; mais deux finalités sont distinguables .

1. L'instruction est utilisée pour la recherche d'un mot quelconque dans une zone (non secrète) afin de lire son contenu .
2. L'instruction est utilisée pour la recherche d'une entête de niveau ou de zone afin de sélectionner ce niveau ou cette zone . (cfr. 6.1.1.4 Le principe d'exploitation des niveaux et des zones) .

L'application doit fournir , avec cette instruction , les paramètres suivants :

- l'adresse du début de la recherche .
- Le sens (croissant ou décroissant) de la recherche
- l'argument : c'est-à-dire le mot (entête) recherché
- le masque à utiliser , ce dernier permettant d'effectuer une recherche sur une partie précise du mot de 32 bits (entête) .

Après l'activation de cette instruction , la carte se contente de renseigner l'application sur le succès ou non de la recherche .

B. Lecture du résultat **(L.RES.)** :

Les finalités de cette instruction varient selon son contexte d'utilisation :

- 1 Suite à une instruction de recherche sur arguments : le résultat envoyé par la carte comprend principalement l'adresse et le contenu du mot (entête) recherché .
- 2 Suite à certaines instructions algorithmiques : le résultat communiqué par la carte représente le résultat de l'exécution de l'algorithme Télépass .
- 3 Suite à une instruction de génération d'un nombre aléatoire (cfr. le point D) : le résultat communiqué par la carte est le nombre aléatoire généré par celle-ci .

C. Lecture de L octets **(LECT.)** :

Cette instruction a pour but de lire un nombre précis d'octets (L) à une adresse-carte spécifiée .

La réussite de cette lecture nécéssite la sélection préalable de la zone et/ou du niveau comprenant cette adresse . La réussite de cette sélection pouvant être tributaire d'une présentation de clé (cfr. l'indicateur P.L. dans l'entête d'une zone de travail) .

Lorsque un niveau est sélectionné : seules les entêtes de zones du niveau courant et les entêtes des niveaux inférieurs sont accessibles en lecture.
Lorsque une zone est sélectionnée: seuls les mots contenus dans cette zone sont accessibles selon les règles de lecture de celle-ci.

D. Génération d'un nombre aléatoire **(RAND.)** :

Cette instruction demande à la carte de fournir à l'application un nombre aléatoire . Elle est surtout activée avant une authentification du porteur , du terminal ou de l'émetteur .

E. Ecriture directe d'un mot **(ECR.)** :

Cette instruction réalise l'écriture du mot à l'adresse communiquée. Cette adresse ne peut être comprise dans une zone secrète , d'accès ou de fabrication et ne peut référencer un mot déjà validé .

Comme nous l'avons vu précédemment, cette instruction est aussi utilisée lors de la création de zones ou de niveaux. (cfr. 6.1.1.4. Le dynamisme des niveaux : la création de zones) .

F. Ecriture d'un mot de test **(E. TST.)** :

Pour mémoire (cfr. 3.2.4.3. Les instructions simples).

G. Ecriture du verrou S **(E. VER.)** :

Le verrou S (présent dans les entêtes d'allocation et de contrôle) indique , une fois positionné à 0 , que le niveau entre dans sa phase active .

6.1.2.3. INSTRUCTIONS BASEES SUR L'ALGORITHME TELEPASS.

A. Authentification du terminal **(AUTH. TERM.)** :

Cette opération est réalisée par la présentation à la carte porteur d'un certificat calculé par le terminal (ou plutôt par sa carte applicative) . Cette authentification n'est provoquée que si le bit T (cfr. l'entête d'allocation et de contrôle) du niveau sélectionné est positionné . Aucune zone d'accès n'est mise à jour lors de cette instruction ; mais la carte peut , le cas échéant (échec) , rester muette .

B. Authentification du porteur par clé porteur **(AUTH. POR. 1.)** :

Cette instruction , visant à authentifier l'utilisateur de la carte porteur peut s'effectuer de deux manières distinctes :

1) avec chiffrement **(AUTH. POR. 1a.)**
2) sans chiffrement **(AUTH. POR. 1b.)**

Dans les deux cas, la zone d'accès du niveau carte est mise à jour et le résultat de cette instruction est gardé en interne dans la carte . L'indicateur CP. (cfr. l'entête de la zone d'accès du niveau carte) limite le nombre d'essais autorisés .

C. Authentification du porteur par clé alternative **(AUTH. POR. 2.)** :

Cette authentification et ses principes sont semblables à ceux vus ci-dessus . Les versions avec ou sans chiffrement existent aussi et l'indicateur CP. joue le même role . La seule différence réside dans le fait qu'une seule clé porteur est active à la fois ; tandis que plusieurs clés alternatives peuvent être utilisables simultanément .

D. Authentification de l'émetteur par clé émetteur **(AUTH. EME.)** :

L'authentification de l'émetteur d'un niveau est éffectuée grâce à cette instruction . La clé utilisée est la clé émetteur et la zone affectée est celle du niveau contenant cette clé . L'indicateur CA. (cfr. l'entête de la zône d'accès du niveau concerné) nous indique le nombre de présentation maximal avant blocage . Si le niveau est bloqué, l'instruction est utilisée dans le cadre d'un déblocage .

E. La téléécriture **(TELEC.)** :

Cette instruction réalise l'écriture à distance de données confidentielles et en même temps la signature électronique .

Via cette instruction , la carte accepte d'écrire dans sa mémoire des mots (secrets , applicatifs ou entêtes) dont elle est sûre de l'origine . L'émetteur doit pour cela chiffrer , avec sa propre clé , les données transmises à la carte porteur . Cette dernière déchiffre ces données , vérifie leur cohérence et les écrit en mémoire .

F. La certification **(CERTIF.)**

La certification est utilisée pour garantir la présence d'informations dans une carte . Elle ne peut concerner un mot protégé tant que l'authentification (préalablement nécessaire) n'a pas réussie . Elle ne peut en aucun cas être éffectuée sur un mot secret . La clé utilisée est la clé de certificat , aucune zone d'accès n'est affectée .

G. Authentification de la carte **(AUTH. CARTE.)**:

Cette instruction est utilisée pour authentifier les cartes porteurs . Cette authentification se résume à demander à la carte porteur de calculer un certificat . La clé utilisée est la clé de certificat , aucune zone d'accès n'est affectée .

H. Authentification du porteur par clé biométrique **(AUTH. POR. 3.)** :

Cette instruction consiste à soumettre à la carte porteur les informations biométriques fournies par l'utilisateur de la carte .

Il est actuellement possible de stocker - sur la carte porteur - une digitalisation des empreintes ou d'une signature ; mais il est encore techniquement impossible de confier à une carte à microprocesseur la comparaison de telles informations . Cette dernière doit donc être réalisée dans le terminal-caisse par un module de comparaison .

La carte porteur confie donc la responsabilité de cette comparaison au terminal-caisse , ce qui nous amène à craindre des tentatives de fraudes . La carte s'en prémunit en n'acceptant , du terminal-caisse , que des réponses résultats de l'algorithme Télépass et chiffrées par la clé biométrique .

C'est la zône d'accès du niveau carte qui est mise à jour . L'indicateur CP. joue le même rôle que lors des authentifications du porteur .

6.2. LES AVANTAGES DU MASQUE M.A. .

Le masque devenant plus puissant , sa relative complexité s'est accrue .

Tout en gardant certains principes , la gestion mémoire s'est modifiée d'ou une évolution des ordres simples . Suite à la diversité des clés pouvant être stockées dans cette mémoire et suite aux modifications apportées à Télépass ; divers ordres algorithmiques sont apparus .

Malgré l'interdépendance entre la gestion de la mémoire et le jeu d'instructions de la carte , ce point se propose de distinguer :

- Les avantages apportés par le jeu d'instructions.
- Les avantages apportés par la gestion des mémoires.

6.2.1. LES AVANTAGES APPORTES PAR LE JEU D INSTRUCTIONS

En toute généralité , les instructions de la carte sont plus conviviales . La carte (excepté dans le cas d'une fraude probable) communique rapidement et avec précision son état suite à la dernière instruction reçue .
Cet interfacage plus évolué est utile pour le programmeur d'application , qui après chaque instruction simple , algorithmique ou mise sous tension peut connaître l'état de la carte (muette , bloquée , ...) .

L'évolution au niveau des instructions algorithmiques est plus sensible qu'au niveau des instructions simples . Après avoir vu quelques instructions simples , nous étudierons les fonctions de sécurité rendues possibles ou modifiées par l'évolution des instructions algorithmiques .

6.2.1.1 LES INSTRUCTIONS SIMPLES.

A. Recherche sur arguments.

Vu la gestion de la mémoire et connaissant le principe d'exploitation des niveaux et des zones , nous comprenons que cette instruction soit vitale (cfr. 6.1.1.4) . En effet , la sélection d'un niveau ou d'une zone est éffectuée grâce à l'ordre de recherche sur arguments , afin de trouver l'entête de la zone ou du niveau concerné .
De plus , grâce à sa paramétrabilité , elle facilite le travail de l'application qui doit à maintes reprises accéder rapidement au contenu de certains mots-carte (Ex. date , identité , plafonds) . Ces mots , de type et de confidentialité différents , sont éparpillés dans diverses zones de la mémoire carte .

B. Lecture de L. octets et Ecriture directe d'un mot.

Suite au principe d'exploitation des niveaux et des zones , les instructions de validation en lecture et en écriture ont disparues (cfr. 3.3.3 Exemples typiques de communication avec une carte). En effet , pour écrire ou lire dans une zone protégée , if faut d'abord la sélectionner et lui présenter sa clé .

La suite d'instructions de sélection combinées aux présentations de clés requises , remplace et simplifie désormais le scénario de lecture et d'écriture dans une zône protégée ou non .

C. Génération d'un nombre aléatoire.

Jusqu'à présent , ce nombre aléatoire - sécurisant les dialogues - était fourni par l'application . La carte (porteur ou applicative) peut maintenant le générer , ce qui sécurise mieux les authentifications et autorise un contrôle de type aléatoire (cfr. Phase 9) .

L'apparition de cette instruction a aussi permis le scénario d'authentification du terminal , elle doit être complètement contrôlée par la carte porteur.

6.2.1.2 LES INSTRUCTIONS ALGORITHMIQUES.

Les avantages des instructions algorithmiques provenant de Télépass , attardons nous un instant sur celui-ci .

Le masque abrite une nouvelle version de Télépass . Les principales modifications sont les suivantes :

1. La fonction Télépass est désormais disponible sous deux versions :

   R = F (E , S , adr. carte, (adr carte))
   E = F' (R ,S , adr. carte , (adr. carte))

   La forme inverse F' est implantée dans la carte et protégée de la même façon que la première version .

2. Il est désormais possible d'activer F (ou F') , en utilisant comme paramètre S : une autre clé que l'ancienne clé interne de la carte (Ex : soit une clé porteur , émetteur , système ou certificat)
   Comme nous allons le constater plus loin : cette amélioration de la souplesse de la fonction Télépass a permis une plus grande variété dans les instructions algorithmiques . Ces instructions participent aux diverses authentifications , certifications , signatures électroniques .

La nouvelle version de Télépass ayant augmenté le nombre d'instructions algorithmiques (cfr. 6.1.2.3 Instructions basées sur l'algorithme Télépass) , voyons l'influence de ces dernières sur les fonctions de sécurité (vues au chapitre 4) et de nouvelles :

- L'authentification du porteur :
  par clé porteur
  par clé alternative
  par clé biométrique
- L'authentification du terminal
- L'authentification de l'émetteur
- La certification et l'authentification de la carte
- La téléécriture

Si l'application se limite à éffectuer une authentification , les paramètres "adr. carte" et "(adr. carte)" sont inutilisés . Dans le cas où l'application réalise en même temps une certification d'un mot carte , ils identifient le mot à utiliser comme un paramètre de Télépass . (Ex. Dans la 6° et 7° phase , le terminal-caisse s'assure que le n° de compte et le dernier crédit disponible a correctement été lu .)

Rem .

Durant les différents dialogues entre la carte porteur et le terminal-caisse , il est utile de se rappeler que dans le terminal est insérée une carte applicative similaire à la carte porteur . Seuls les dialogues de l'application avec la carte porteur sont détaillés .

A. Authentification du porteur.

Par clé porteur.

1. avec chiffrement .

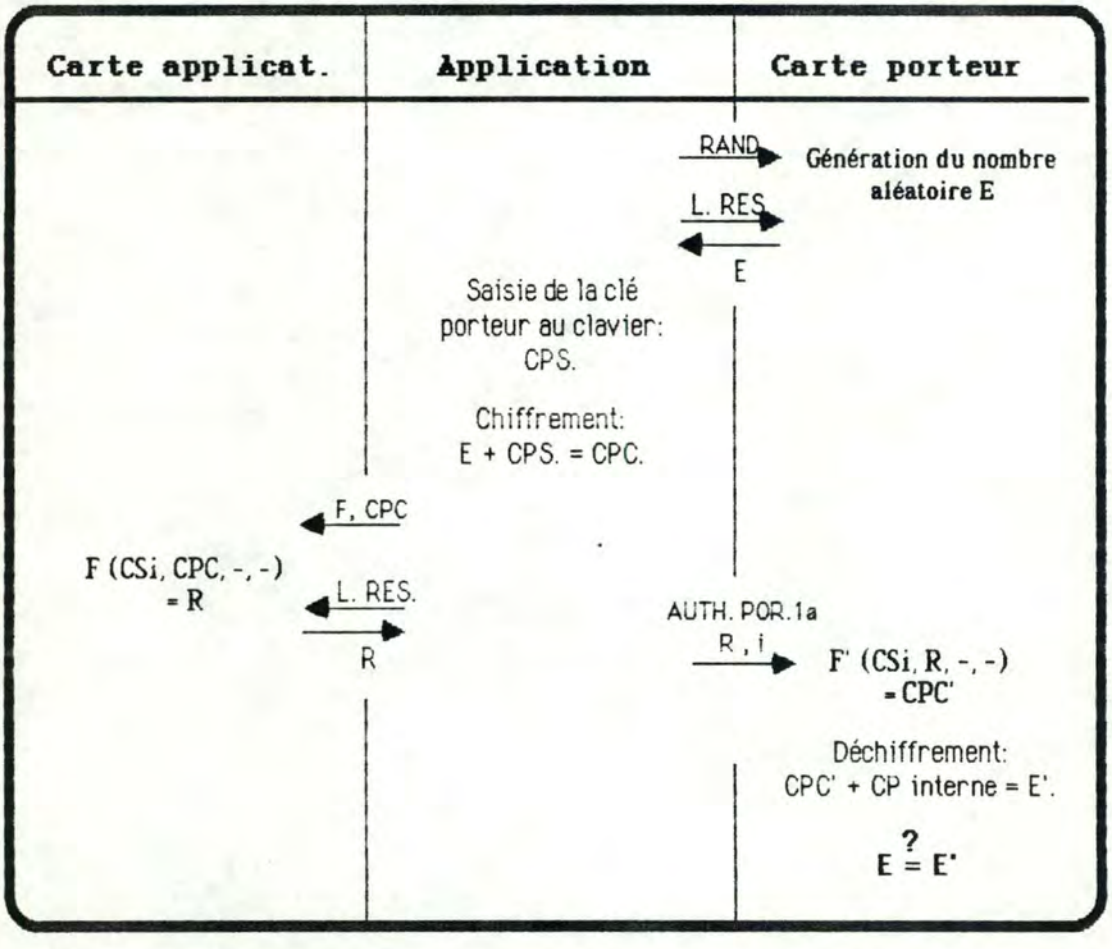


Fig. 6.13 : Auth. porteur par clé porteur (avec chiffrement) .

- L'application demande à la carte porteur de générer (RAND.) un nombre aléatoire (E) qu'elle lit par la suite (L. RES.) .
- L'application saisit la clé porteur au clavier du client (CPS.)
- Chiffrement : l'application exécute un ou-exclusif avec la clé porteur saisie (C.P. saisi) et la nombre aléatoire reçu (E) . Le résultat est noté C.P.C. .
- L'application demande à la carte applicative d'activer F en prenant comme paramètres :

  - une clé système identifiée par i (C.S.i)
  - la clé porteur chiffrée (C.P.C.)

- Le résultat R est lu par l'application (L. RES.) .
- L'application demande à la carte porteur d'activer F' (AUTH. POR. 1a) , donc de calculer un certificat (C.P.C') en prenant comme paramètres :

    - une clé système identifiée par la valeur i reçue(C.S.i)
    - le résultat R reçu .

- Déchiffrement : la carte porteur exécute un ou-exclusif avec le résultat C.P.C' et la clé porteur de la carte (C.P.I.) .
- Le résultat E' doit être semblable au nombre aléatoire généré plus haut par la carte . Dans ce cas , le porteur est authentifié .

2. sans chiffrement .

Ce scénario d'authentification du porteur ignore les opérations comprises entre le chiffrement et le déchiffrement . Le résultat issu du chiffrement est directement envoyé par l'application vers la carte porteur(AUTH. POR. 1b) qui le déchiffre et teste le résultat .

Par clé alternative.

Le scénario d'authentification du porteur par clé alternative est similaire à celui par clé porteur . Il n'est plus demandé à l'utilisateur de la carte de taper sa clé porteur , mais la réponse à une autre question (Ex. date de naissance) . Cette réponse est la clé alternative (à la clé porteur) , elle identifie l'utilisateur .

Il suffit donc pour le lecteur de remplacer dans les scénarios (avec et sans chiffrement) les occurences de "clé porteur" par celles de "clé alternative" . Les fonctions activées dans la carte applicative et dans la carte porteursont respectivement :

F (CSi , C Alt. , - , -) et F' (Csi , C Alt. , - , -) .

La clé alternative pourrait exister en multiples versions , elle serait dès lors identifiée par j (C Alt. j) .

Par clé biométrique.

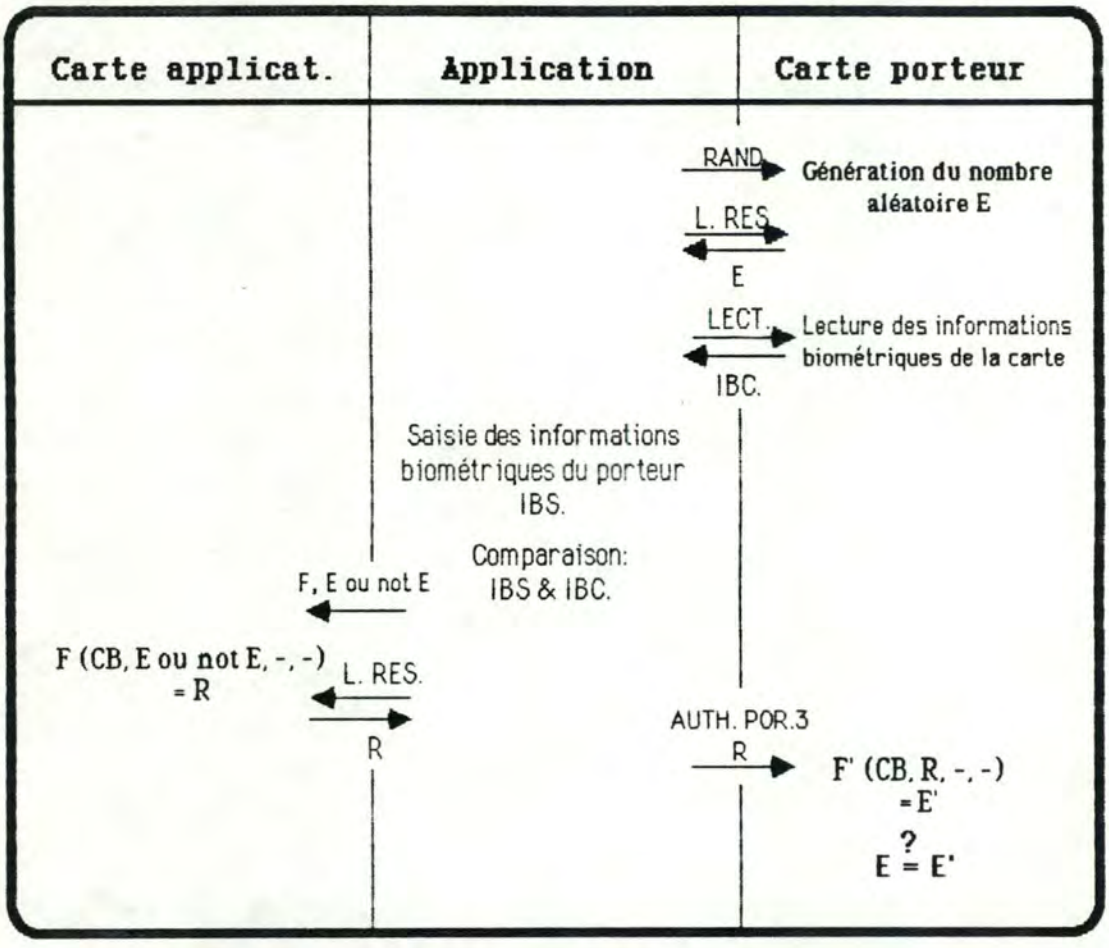


Fig. 6.14 : Auth. porteur par clé biométrique .

- L'application demande à la carte porteur de générer (RAND.) un nombre aléatoire (E) qu'elle lit par la suite (L. RES.) .
- L'application lit les informations biométriques contenues dans la carte (IBC.)
- L'application saisit les informations biométriques identifiant le porteur de la carte (IBS.)
- L'application compare - via un module de comparaison - les deux informations biométriques (IBS. et IBC.)
- L'application demande à la carte applicative d'activer F en prenant comme paramètres :

  - la clé biométrique (C.B.)
  - soit le nombre aléatoire reçu (E) : dans le cas d'une similitude entre IBS. et IBC.
  - soit tout nombre aléatoire différent de E : dans le cas d'une non-similitude entre IBS. et IBC.

- Rem. : l'application décide délibérément de communiquer ou non , à la carte applicative , le nombre aléatoire généré par la carte porteur . En effet , si les signatures ou les empreintes ne correspondent pas , l'application communique un nombre quelconque différent de E . La carte porteur concluera donc par un échec de l'authentification . Dans le cas d'une similitude , le nombre E est envoyé normalement à la carte applicative .
- Le résultat R est lu par l'application (L. RES.) .
- L'application demande à la carte porteur d'activer F' (AUTH. POR. 3) , donc de calculer un certificat (E') en prenant comme paramètres :

  - la clé biométrique (C.B.)
  - le résultat R reçu .

- Le résultat E' doit-être semblable au nombre aléatoire généré plus haut par la carte . Dans ce cas , le porteur est authentifié .

B. Authentification du terminal.

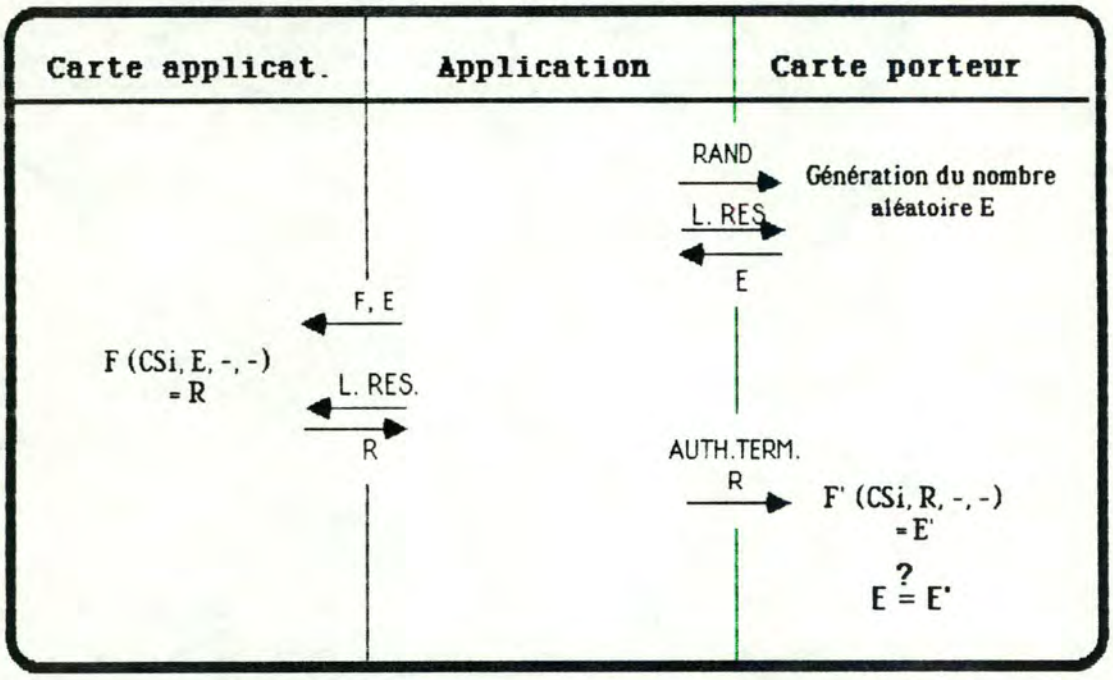


Fig. 6.15 : Authentification terminal .

- L'application demande à la carte porteur de générer (RAND.) un nombre aléatoire (E) qu'elle lit par la suite (L. RES.) .
- L'application demande à la carte applicative d'activer F en prenant comme paramètres :

    - une clé système identifiée par i (C.S.i)
    - le nombre aléatoire reçu (E)

- Le résultat R est lu par l'application (L. RES.) .
- L'application demande à la carte porteur d'activer F' (AUTH. TERM.) , donc de calculer un certificat (E') en prenant comme paramètres :

    - une clé système identifiée par la valeur i reçue (C.S.i)
    - le résultat R reçu .

- Le résultat E' doit être semblable au nombre aléatoire généré plus haut par la carte . Dans ce cas , le terminal est authentifié .

C. Authentification de l'émetteur.

Le scénario de cette authentification est identique à celui d'une authentification de terminal . Le rôle joué par le terminal-caisse peut l'être par tout autre système informatique possédant un module sécuritaire .

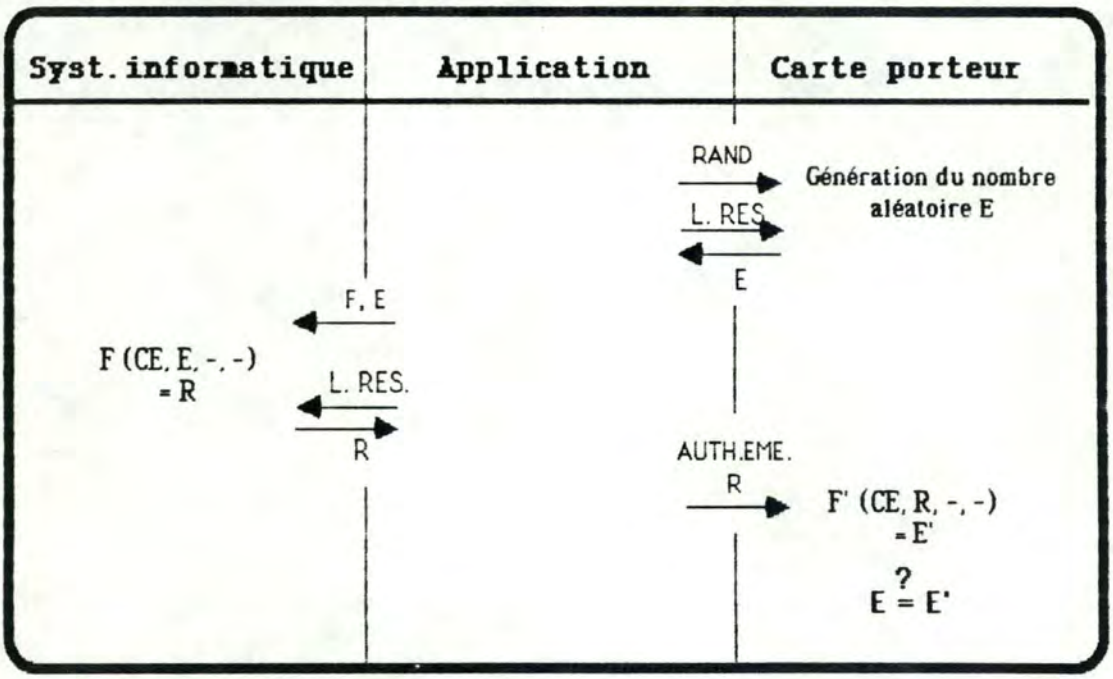


Fig. 6.16 : Authentification émetteur .

- L'application demande à la carte porteur de générer (RAND.) un nombre aléatoire (E) qu'elle lit par la suite (L. RES.) .
- L'application demande au système informatique d'activer F en prenant comme paramètres :

  - la clé émetteur (C.E.)
  - le nombre aléatoire reçu (E)

- Le résultat R est lu par l'application (L. RES.) .
- L'application demande à la carte porteur d'activer F' (AUTH. EME.) , donc de calculer un certificat (E') en prenant comme paramètres :

  - la clé émetteur (C.E.)
  - le résultat R reçu .

- Le résultat E' doit être semblable au nombre aléatoire généré plus haut par la carte . Dans ce cas , l'émetteur est authentifié .

D. Certification et Authentification de la carte.

Le scénario d'une demande de certificat ne varie que sensiblement de celui vu lors de l'implémentation des fonctions de sécurité (cfr. 4.1.2.1 Calcul de certificat et authentification d'une carte)

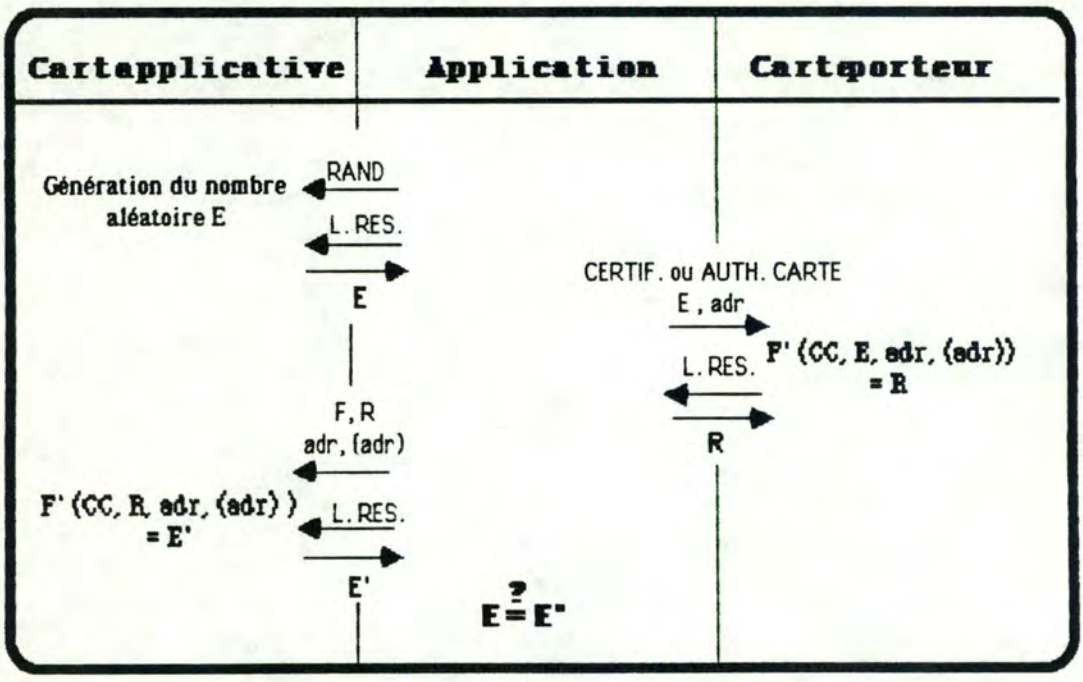


Fig. 6.17 : Certification et Authentification de la carte .

- L'application demande à la carte applicative de générer (RAND.) un nombre aléatoire (E) qu'elle lit par la suite (L. RES.) .
- L'application demande à la carte porteur d'activer F (CERTIF. ou AUTH.) en prenant comme paramètres :

    - la clé de certificat (C.C.)
    - le nombre aléatoire reçu (E)
    - l'adresse fournie (ADR.)
    - le contenu de cette adresse ((ADR.))

  Ces deux derniers paramètres sont quelconques lors d'une authentification de carte mais pointent vers un mot carte lors d'une certification .
- Le résultat R est lu par l'application (L. RES.) .
- L'application demande à la carte applicative d'activer F' , donc de calculer un certificat E' en prenant comme paramètres :

    - la clé de certificat (C.C.)
    - le résultat reçu (R)
    - l'adresse reçue (ADR.)
    - le contenu de cette adresse ((ADR.))

- Le résultat E' est lu par l'application (L. RES.) .
- L'application compare ce nombre aléatoire E' avec le premier nombre aléatoire E - généré plus haut par la carte applicative -. Dans le cas d'une similitude , la certification ou l'authentification de la carte est réussie .

E. Téléécriture.

La téléécriture sécurise de façon maximale l'écriture :

- des mots secrets (Ex. clé porteur , clé système)
- des mots applicatifs (Ex. nouveau crédit disponible , plafonds ....)
- d'entêtes (Ex. nouvelles zones , niveaux)

Tous les traitements annexes au paiement de contact , peuvent être réalisés auprès de n'importe quel terminal en connexion avec un centre disposant de la clé émetteur et de l'ordre de téléécriture .

Rem. : le déchiffrement d'informations est désormais rendu possible grâce à la détention par les cartes des deux versions de Télépass (F et F') . Le scénario serait plus ou moins semblable à celui d'une téléécriture .

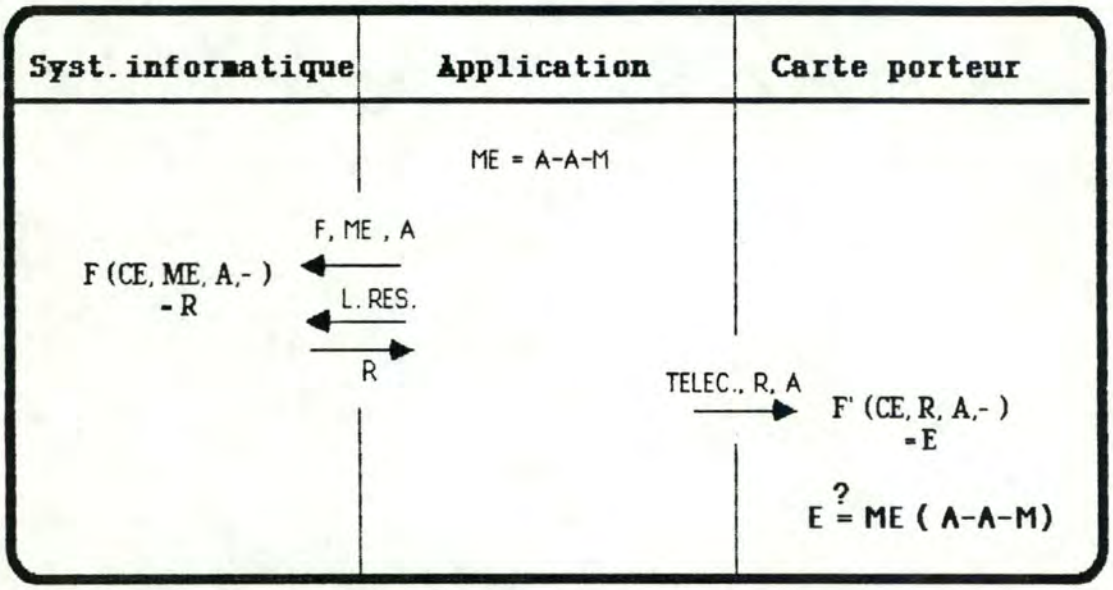


Fig. 6.18 : Téléécriture .

- L'application compose le message ME. , en plaçant la double adresse A dans sa première partie (A étant l'adresse ou le mot doit être écrit) . La seconde partie de ME. contient le mot M à écrire .
- L'application demande au système informatique d'activer F en prenant comme paramètres :

  - la clé émetteur (C.E.)
  - le message reçu (ME.)
  - l'adresse reçue (A.)
  - un mot vierge (-)

- Le résultat R est lu par l'application (L. RES.) .
- L'application demande à la carte porteur d'activer F' (TELEC.) , donc de calculer un certificat E en prenant comme paramètres :

  - la clé émetteur(C.E.)
  - le résultat reçu (R)
  - l'adresse reçue (A.)
  - un mot vierge (-)

- La cohérence de E est vérifiée de la manière suivante : si le résultat E contient dans sa première partie une double version de l'adresse reçue lors de l'ordre de téléécriture (A.) ; la seconde partie de E contenant le mot M est écrite à l'adresse A .

6.2.2. LES AVANTAGES APPORTES PAR LA GESTION DE MEMOIRE

Les deux aspects les plus importants , abordés ci-dessous , concernent la polyvalence et le caractère multi-services de la carte .

Tout au long de ces deux points , le lecteur pourra constater une grande souplesse dans la personnalisation et l'utilisation d'une telle carte . Les possibilités de paramétrage y sont pour beaucoup .

6.2.2.1 LA POLYVALENCE.

La carte à microprocesseur est appelée à être utilisée dans divers contextes d'utilisations . Cette polyvalence est d'autant plus nécessaire lorsque l'on sait qu'une carte applicative n'est qu'une carte à microprocesseur dont la fonction est différente .

C'est au travers des niveaux , des zones et de leur contenu que nous allons étudier cette polyvalence . Elle sera illustrée plus loin au travers de l'application bancaire étudiée .

A. Niveaux.

Une application peut désormais bénéficier du principe de hiérarchisation de la structure des "fichiers d'une carte" . Elle utilisera la carte dans une situation plus complexe (Ex. Un client disposant de deux comptes auprès d'une banque , peut choisir - lors de ces achats - le compte à débiter) .

B. Zones.

La présence d'un type de zone dans un niveau reste facultative . A l'inverse les zones peuvent exister en plusieurs exemplaires à l'intérieur d'un même niveau .

C. Contenu.

L'émetteur est libre d'enregistrer , dans les zones de travail et publique , des informations de nature et de protection diverses . En effet , aucune contraintes sémantiques n'est imposée sur leur contenu .

Zone secrète : grâce à la notion de version et de multiplicité des clés , une application peut enregistrer , dans une zone secrète , un nombre indéterminé de clés de toute sorte .

Zone publique : elle accepte tout type d'informations en provenance de l'émetteur .

Zone travail : cette zone ,acceptant toutes les informations issues de l'application ; il est aisé d'implémenter , par exemple l'option de budget de l'application de paiement (cfr. chapitre 5) . La carte doit pour cela , uniquement stocker un identifiant "budget" pour toutes les transactions dans sa mémoire de travail.

La polyvalence se caractérise par la capacité de la carte à contenir tous types d'informations . Leur organisation doit être flexible et toute application doit être possible à partir du jeu d'instructions disponible .

En ce qui concerne les instructions , seule l'apparition de l'instruction d'effacement (garantie au maximum) est attendue avec impatience . Celle-ci devra adopter les principes utilisés dans une écriture ou une lecture (les protections , ...) .

En ce qui concerne les informations et leur organisation , c'est au travers des notions de niveaux , zones et contenu de mots que nous voudrions illustrer les choix que peut effectuer un émetteur lors de la personnalisation de ses cartes .

D. Illustration.

Cette polyvalence est illustrée , en partie , par la configuration mémoire choisie pour l'application du paiement de contact .

ZONE FABRICATION

(N. CARTE)

ZONES SECRETES

(N. CARTE)

ENTETE D'ALLOCATION DE CONTROLE (N. CARTE)

ZONE D'ACCES

(N. CARTE)

ZONES SECRETES

(N. CARTE)

ZONES PUBLIQUES

(N. CARTE)

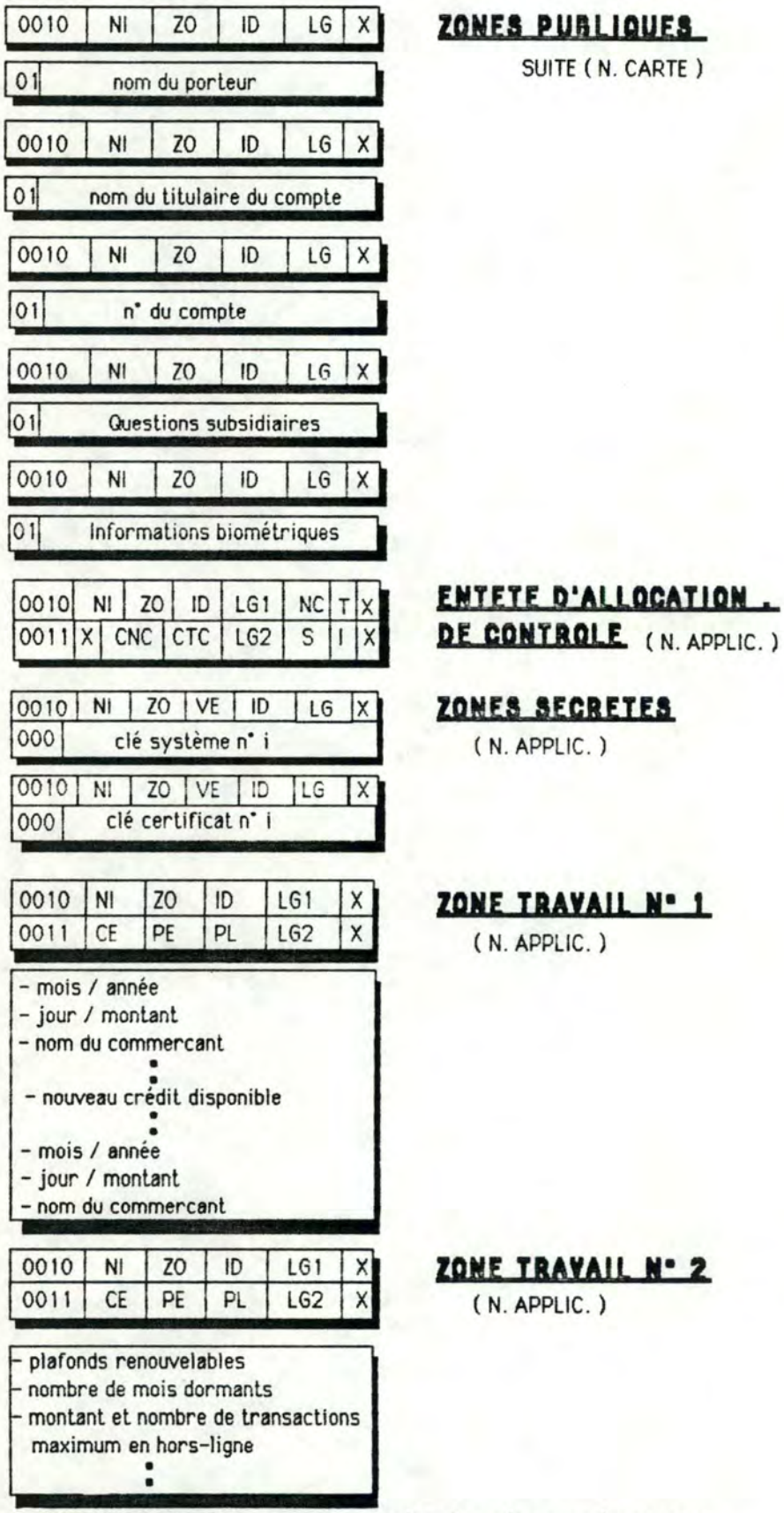


Fig. 6.19 : Configuration mémoire du paiement de contact .

Cette carte possède désormais suffisament d'informations utiles et diversifiées pour effectuer une transaction hors-ligne hautement sécurisée . Ceci étant lié au fait que les zones de travail et secrètes peuvent contenir des plafonds , clés et paramètres facilement renouvelables et que l'application (via le jeu d'instructions) les utilise de façon optimale .

6.2.2.2 MULTI-SERVICES.

L'organisation de la mémoire PROM (cfr. 6.1.1.) , nous illustre la capacité de la carte à abriter divers espaces . D'où la **multiplicité des services** offerts , par la carte , au porteur . (Ex. carte de paiement , de téléphone , de cantine ...) .

Ex. Lors de la phase 2 : la carte porteur abritant diverses applications et le terminal-caisse possédant les divers programmes applicatifs associés : il est demandé au client de déterminer quelle application utiliser pour le paiement de la transaction courante .

Outre cette souplesse ; considérons **l'aspect dynamique** de la création des niveaux et des zones :
Chaque application (niveau carte et application) reçoit lors de sa personnalisation le droit de créer de nouvelles zones , d'étendre son espace et inversément d'en libérer .

Ex. Durant la phase active d'une application , l'émetteur est libre de créer un ou plusieurs services supplémentaires dont il est le responsable . Lors de la phase 9 : la gestion dynamique est utile lorsque le programme applicatif est prévenu que la carte possède une zone d'accès ou de travail saturée . L'application peut étendre le niveau concerné et créer de nouvelles zones de travail ou d'accès .

Enfin , dans un proche avenir , la mémoire pourrait devenir effaçable sélectivement . Ce qui autoriserait une application à vider sa zone de travail ou d'accès dès qu'elle est saturée .

Cette configuration et cette gestion des mémoires doit comprendre une **indépendance** entre les applications ou services . Ceci sous-entend que :

- chaque application ou service est capable d'identifier son propre émetteur via sa clé .

- chaque application ou service est capable d'identifier le terminal-caisse dans lequel la carte est insérée .

- l'identification du porteur est globale : chaque accès par clé porteur met à jour la zone d'accès du niveau carte . Ce choix semble être le bon puisque la carte n'est destinée qu'à un seul utilisateur .

Cette indépendance est gérée grâce à la mémoire de contrôle : elle constitue un élément essentiel du système anti-fraude . Elle est réalisée par l'invalidation ou le blocage .

Ex. En examinant les phases 4,5,7,17 ; nous constatons que la carte doit à certains moments être totalement ou partiellement invalidée et à d'autres totalement ou partiellement bloquée .

Le principe selon lequel chaque niveau peut posséder son propre système de contrôle nous permet de réaliser les deux types de blocage en présentant trois clés fausses au système de contrôle concerné (le niveau carte bloquera la carte complètement , les autres niveaux ne la bloqueront que partiellement) . Le blocage et le déblocage sont gérés de façon indépendante par les niveaux (cfr. les indicateurs CP , CA , CD des entêtes de zones d'accès de chaque niveau) .

L'invalidation est réalisée par le positionnement du bit I présent dans l'entête d'allocation et de contrôle de chaque niveau , ce qui autorise l'invalidation globale ou partielle .

L'invalidation et le blocage partiel ne touche donc qu'un niveau (application ou service) à la fois .

6.3. CONCLUSIONS .

Au cours de ce chapitre , nous avons principalement étudié une gestion mémoire et les instructions d'un nouveau masque (M.A.) .

Les différentes combinaisons niveaux-zones et le libre contenu de celles-ci nous permettent de mieux envisager une proche polyvalence alliée à une souplesse d'utilisation de la carte à microprocesseur . Le caractère multiapplicatif est désormais facilement concevable .

Le nouveau jeu d'instructions du microprocesseur (simples ou basées sur l'algorithme Télépass) nous a permis de réaliser des scénarios auparavant impossibles .

Le masque M.A. semble donc pouvoir offrir au monde bancaire une solution agréable pour son application de paiement de contact . Grâce à sa souplesse , le dossier portable , le contrôle d'accès et la sécurité de l'information pourront facilement et sûrement être développés . La seule entrave à la réalisation d'un tel masque semble être la taille mémoire dont la carte doit disposer pour le stocker .

CHAPITRE 7 :
LE MATERIEL

VII. LA SECURITE DU MATERIEL

La carte porteur , grâce à son autonomie de décision et aux informations détenues , sécurise une bonne partie de l'application bancaire étudiée (cfr . Chapitre 6) .

Le programme applicatif prend aussi en charge des contrôles utiles pour parer les fraudes éventuelles (cfr. Chapitre 5) .

Comme nous allons le constater par la suite : le système complet associé à la carte doit être renforcé , car la sécurité apportée par la carte porteur et l'application est insuffisante .

Ce chapitre traitera de la sécurité du terminal-caisse , de la carte applicative et enfin de leur coopération.

7.1 LE TERMINAL-CAISSE

Le système terminal-caisse vu au point 5.2.1 (La configuration du système) est utilisé dans les transactions financières (en-ligne ou hors-ligne). Il approuve ou non une transaction en se basant sur les informations détenues par lui-même , par la carte porteur et parfois celles recues lors d'une connexion en-ligne avec le central .

Etant donc amené à prendre des décisions importantes et manipulant des données confidentielles , il est normal de songer à sa protection .

Après avoir fait une rapide description du système informatique qu'est le terminal-caisse et après avoir énuméré ses diverses fonctions , nous passerons en revue les éléments du terminal-caisse devant être particulièrement protégés .

7.1.1. LE SYSTEME INFORMATIQUE .

Le terminal-caisse est appelé à gérer les divers éléments de son environnement ; il dispose , pour cela , d'un processeur central et de mémoires . Il est l'équivalent d'un microordinateur .

7.1.1.1 LE PROCESSEUR CENTRAL

Le processeur régit le contrôle de tous les éléments présents dans l'environnement du terminal-caisse . Il peut activer toutes les fonctions relatives à sa propre gestion et celles des périphériques . Des processeurs standards du marché peuvent-être utilisés (8035 , 8036 ....) .

7.1.1.2 LES MEMOIRES

Divers types de mémoire sont utilisés : RAM , ROM , EEPROM . Elles sont utilisées de la manière suivante :

- une mémoire RAM. contenant les données de travail . Les applications y sont aussi téléchargées . Elle a une capacité de 32 Kbits, pouvant ainsi abriter jusqu'à 8 applications de 4 Kbits .
- une mémoire ROM. contenant l'interpréteur (cfr. 7.1.2.9) .
- une mémoire ROM. contenant l'interfacage avec les éléments du système .
- une mémoire ROM. contenant les routines de tests destinées à vérifier l'état des différents éléments du système .
- une mémoire supplémentaire (EEPROM. ou RAM. sauvegardée) destinnée à stocker les données hors-ligne . Cette mémoire est capable de stocker plus de 250 transactions hors-ligne .

7.1.2. LES FONCTIONS DU MATERIEL .

Les fonctions exécutées par le matériel , y-compris celles relatives à la transaction monétaire , sont les suivantes :

- installation du terminal-caisse .
- mise à jour de l'horloge et de paramètres de fonctionnement .
- gestion des échanges avec les éléments "externes" du teminal-caisse .
- gestion des lots de transactions .
- gestion d'un protocole d'échange de bout en bout avec l'application distante .
- gestion des incidents .
- gestion de programmes d'auto-test .
- téléchargement de logiciels et de paramètres .
- interpréteur .

7.1.2.1 INSTALLATION DU TERMINAL-CAISSE

L'installation est exécutée lors de chaque démarrage et consiste pour le terminal-caisse à acquérir , à partir des diverses cartes applicatives , les données nécessaires au bon déroulement des transactions :

- nom et raison social du commerçant ,
- date , heure ,
- liste des cartes et émetteurs en opposition ,
- logiciels d'application .

7.1.2.2 MISE A JOUR

Les diverses informations initialisées ci-dessus peuvent être mise à jour . L'horloge est mise à jour lors de chaque transmission avec le central , ainsi que les autres paramètres .

7.1.2.3 GESTION DES ECHANGES

Le protocole d'échange avec les cartes (porteur ou applicative) reste conforme aux normes ISO. sur les cartes à microprocesseur (cfr. 3.2.5) .
Le terminal-caisse doit aussi gérer le protocole d'échange avec les autres éléments (imprimantes , claviers , écrans ...) .

7.1.2.4 GESTION DES LOTS DE TRANSACTIONS

Les transactions sont enregistrées dans la mémoire PROM. du terminal-caisse pour être transmises ultérieurement vers un centre. Le lot de transactions est composé d'une suite d'enregistrements . Chaque enregistrement comprend :

- un identifiant de l'application choisie
- une référence des comptes bancaires à créditer et à débiter
- la transaction

L'ouverture d'un lot consiste en une mise à zéro des compteurs . La fermeture totalise le nombre d'enregistrements , le montant total brut du lot de transactions et édite un ticket justificatif .

7.1.2.5 GESTION D'UN PROTOCOLE DE BOUT EN BOUT

Le terminal-caisse est doté d'un modem téléphonique à numérotation automatique , en vue de satisfaire les éventuelles demandes d'autorisation au central . Les données transmises sont indépendantes du protocole choisi pour la communication .

7.1.2.6 GESTION DES INCIDENTS

Suite à une micro-coupure de courant , l'opération doit être reprise à son dernier état et mené à bonne fin . En cas de coupure prolongée : si la clé porteur a déjà été introduite , la transaction doit être terminée ; sinon elle est annulée .

7.1.2.7 PROGRAMMES D'AUTO-TEST

Une série de tests concernant tous les éléments du système est provoquée à la mise sous tension du terminal-caisse . Les tests spécifiques sont déclenchés lors d'une détection d'erreurs . En cas de problèmes , le commercant en est averti .

Suivant l'élément touché et la gravité de la panne , le terminal-caisse peut devenir inutilisable .

7.1.2.8 TELECHARGEMENT

En dehors de l'installation , un logiciel d'application en provenance du central ou de la carte applicative peut être téléchargé dans une mémoire du terminal-caisse . Il en est de même pour diverses tables , paramètres et listes.

Les altérations intentionnelles ou accidentelles doivent être évitées soit par un téléchargement chiffré ou un téléchargement bloc par bloc comprenant une signature à chaque bloc .

7.1.2.9 INTERPRETEUR

Les programmes applicatifs étant écrits dans un langage interpreté , ils font usage de cet interpréteur . Les applications sont dès lors interprètées en ordre élémentaires compréhensibles par le processeur central du terminal-caisse .

7.1.3. LA SECURITE

La liste suivante illustre les éléments du système dont la fiabilité doit-être renforcée au maximun . Une défaillance de ces éléments remettrait en cause la validité des informations écrites dans la carte ou stockées dans la mémoire du terminal-caisse .

- L'affichage
- L'horodateur
- L'alimentation de secours
- Les mémoires
- Divers

7.1.3.1 L'AFFICHAGE

Le terminal-caisse doit conserver la maîtrise exclusive de l'écran pour les dialogues avec l'usager . Cet affichage destinné au client doit être contrôlé rigoureusement car il faut éviter que le client ne délivre sa clé porteur à un terminal trafiqué .

Ex . Il suffirait , suite à un faux message de demande d'introduction de clé porteur , d'intercepter la clé présentée .

De même , l'affichage du montant de la transaction doit être contrôlé .

7.1.3.2 L'HORODATEUR

L'importance de la date fournie par le terminal-caisse (Ex. 3° Phase) est compréhensible lors du contrôle de validité des dates .

De plus , l'écriture d'une date incorrecte pour une transaction fausserait les contrôles effectués lors de la transaction suivante (cfr. Phase 9 : Le contrôle des plafonds) .

7.1.3.3 L'ALIMENTATION DE SECOURS

Le matériel est soumis aux contraintes de son environnement tel que les perturbations du réseau électrique . De ce fait , aucune opération ne doit être entamée sans que l'énergie de secours disponible ne permette son achèvement . (cfr. Phase 10 et 18) .

Toutefois , l'opération peut être annulée automatiquement tant que la clé porteur n'a pas été saisie . Au-delà de cette phase (8°) , l'opération doit se dérouler normalement .

7.1.3.4 LES MEMOIRES

Le terminal-caisse est un organe de décisions basées sur des données et sur l'application à exécuter .

Ces données , applications et décisions sont représentées dans les mémoires du terminal-caisse ; il est donc important d'assurer leur protection .

Cette protection concerne les données de travail (RAM.) , les lots de transactions (ROM.) et les programmes des applications (EPROM.)

A: Les données de travail

Les données ne peuvent être lues ou modifiées sans un contrôle rigoureux sous peine de fausser un contrôle ou encore de réveler des informations secrètes . Elles sont présentes dans des mémoires volatiles , leur origines (ou destinations) sont diverses :

- Carte porteur : n° de compte et nom du client , plafonds .....
- Carte applicative : tables , plafonds .....
- Claviers : clé porteur , montant , type .....
- Central : cfr. phase 17 et 16

B: L'enregistrement des transations

Ces enregistrements stockés , lors d'une transaction hors-ligne , dans une mémoire du terminal-caisse et transférés plus tard vers le central doivent être protégés (cr. Phase 12) .

C: L'application

Le logiciel d'application étant téléchargé , puis stocké dans le terminal-caisse , il est nécessaire de se prémunir contre les modifications ou les altérations accidentelles .

7.1.3.5 DIVERS

La clé porteur : elle est composée par le client puis transférée dans le terminal-caisse . Mis à part la protection de la mémoire réceptrice , il est nécessaire de songer à celle du cable reliant le clavier au terminal . Une configuration monobloc du terminal-caisse avec le clavier du client est aussi envisageable .

Les connexions pirates : afin de parer la connexion au terminal-caisse d'un micro-ordinateur , ce dernier voulant simuler le comportement de la carte porteur ou applicative , une parade semble être le dispositif à avalement des cartes .

7.2 LA CARTE APPLICATIVE

La carte applicative n'est qu'une carte à microprocesseur spéciale dont le rôle est distinct de celle d'une "simple" carte porteur .

Elle est personnalisée et distribuée , par l'émetteur , au commerçant dont le terminal-caisse peut en contenir jusqu'à huit .

Sa protection est compréhensible , lorsque l'on se rappelle brièvement qu'elle est destinée , dans le paiement , à réaliser les fonctions sécuritaires du terminal-caisse et à stocker diverses informations.

Le rôle d'une carte applicative étant différent de celui d'une carte porteur , examinons les fonctions des cartes applicatives et le contenu des mémoires PROM. et EPROM . .

7.2.1. LES FONCTIONS.

La carte applicative dispose sensiblement du même jeu d'instructions qu'une simple carte à microprocesseur . Tous les algorithmes et clés sont implémentés dans les cartes de la même façon que sur la carte à microprocesseur du client (ils ne peuvent 'sortir' de cette carte) .

Grâce à ces instructions , elle doit réaliser les fonctions suivantes :

- stockage d'informations monétaires : taux d'échanges , plafonds,
- stockage d'informations non monétaires : n° de tel. du central ,
- chargement , déchargement et stockage d'applications ,
- exécution d'opérations spécifiques aux applications : auth. , certif. , chiffr. ,
- diversification de clés .

7.2.2. LES MEMOIRES PROM. ET EPROM.

Mis à part une mémoire PROM. similaire à celle vue pour une carte à microprocesseur . La carte applicative peut posseder des mémoires supplémentaires . Ce sont des mémoires EPROM. connectées à la puce vue au point 3.1 . Elles contiennent des informations telles que le programme spécifique à l'application et des données applicatives . Cette mémoire secondaire est adressée de la même manière que la mémoire PROM de la carte .

Cette carte utilise les mêmes principes de gestion mémoire que ceux vus au chapitre 7 . Le contenu des zones secrète , publique et de travail est illustré ci-dessous :

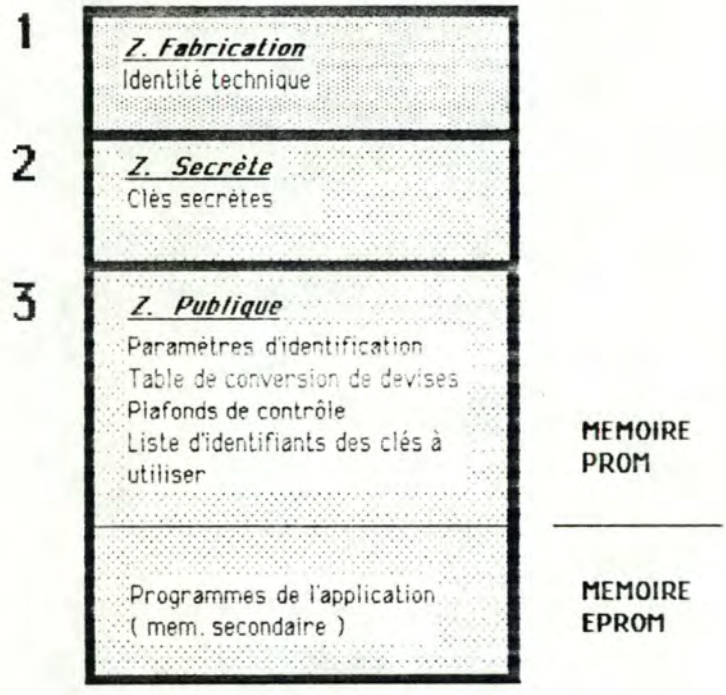


Fig. 7.1 Mémoires PROM. et EPROM.

7.3 LA COOPERATION CARTE APPLICATIVE - TERMINAL

Pour une meilleure compréhension de cette coopération et de la localisation des données détenues par ces deux éléments , illustrons les principaux dialogues entre le terminal-caisse , une carte applicative et le central lors de l'application de paiement de contact .

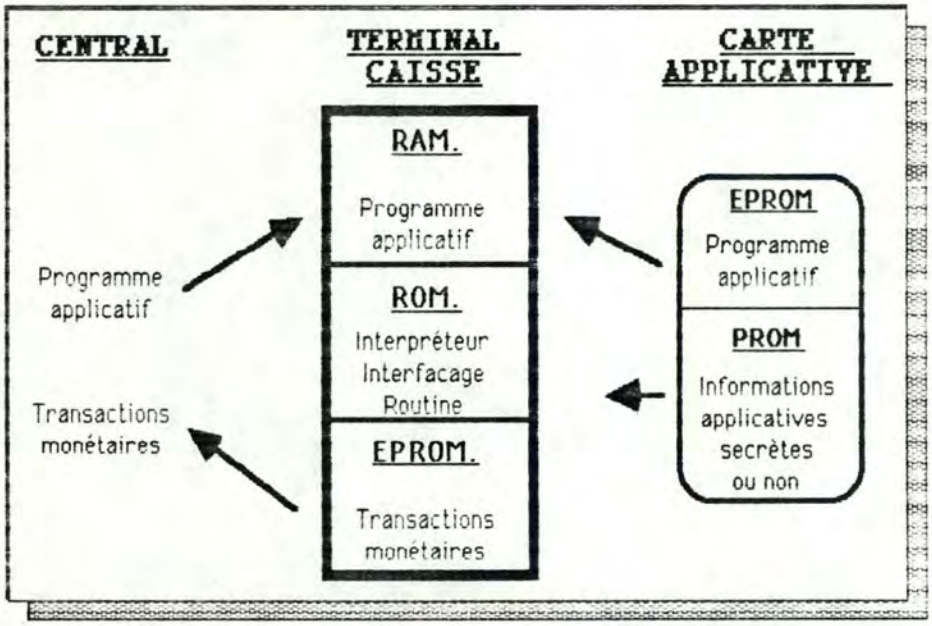


Fig. 7.2 Coopération carte applicative - terminal.

- Le programme applicatif peut être téléchargé de deux manieres distinctes , soit à partir d'une carte applicative contenant cette application dans sa mémoire EPROM , soit à partir du central , via un réseau .
- Une fois le programme applicatif présent dans la mémoire RAM. du terminal-caisse , il est nécessaire de lancer l'interpréteur (ROM.) de ce dernier pour exécuter l'application .
- Les authentifications et les certifications , gérées par le terminal-caisse , sont réalisées par l'intermédiaire de la carte applicative qui détient secrètement la fonction Télépass et les clés secrètes (Ex. contrôler l'authenticité des cartes porteurs , prouver celle des terminaux , certifier) .
- L'accord du terminal-caisse amène celui-ci à devoir lire des informations détenues dans la carte applicative concernée .
- Une fois la transaction acceptée , elle est stockée dans une mémoire EPROM du terminal-caisse , pour être ultérieurement transmise vers le central .

7.4 CONCLUSIONS

Nous pouvons constater que la carte applicative est capable de protéger ses informations et de réaliser les opérations qui lui sont demandées lors des dialogues entre le terminal-caisse et les diverses cartes à microprocesseur . Ceci , grâce aux propriétés intrinsèques d'une carte à microprocesseur .

Le terminal-caisse est aussi amené à devoir protéger ses informations et ses dialogues avec ses périphériques . La plupart des fonctions réalisées par ce terminal-caisse doivent être sécurisées . Le terminal-caisse peut , pour certaines d'entre elles , faire appel à la carte applicative , mais il doit se protéger davantage . Ce point fera l'objet du chapitre suivant .

CHAPITRE 8 :
LE LANGAGE

8.7. EVALUATION

8.7.1. La compacité
8.7.2. Le téléchargement
8.7.3. La sécurité
8.7.4. La portabilité
8.7.5. Interfacage évolué
8.7.6. Evolutivité
8.7.7. Performance

VIII. LE LANGAGE

8.1 INTRODUCTION .

Ce chapitre complète l'étude du système bancaire utilisant la carte à microprocesseur dans le contexte du paiement électronique . Ce dernier maillon concerne le langage utilisé pour développer les applications résidentes ou téléchargées dans les mémoires du terminal-caisse .

Ce langage est un macro-langage orienté carte à microprocesseur .

L'organisation des mémoires du terminal-caisse et la gestion de celles-ci constituent la macromachine .

L'application est développée dans ce langage autour d'une machine virtuelle (la macromachine) , elle est rendue compréhensible au processeur du terminal-caisse grâce à la présence d'un interpréteur dans ce dernier . Ce langage se nomme SCIL pour 'Smart Card Interpreted language' et est en cours de développement à la société Bull CP8 .

Ce chapitre traitera de ce langage et de cette macromachine . Nous comprenons leur nécessité d'être à la fois : sécuritaires, performants et souples pour celui qui est amené à réaliser une application-carte sur un terminal-caisse .

8.2 LES BUTS ET LES MOYENS .

Les buts poursuivis lors de la conception d'un tel système sont divers :

sécurité : nécessité de développer autour de la carte un environnement (le langage et sa macromachine) hautement sécurisé ; ceci pour pallier aux différentes insuffisances de la carte et de l'application (cfr. Chapitre 7 La sécurité du matériel).

portabilité : offrir aux applications , développées sur un hardware précis , la possibilité d'être implantées sur d'autres machines . Une relative indépendance vis à vis de l'évolution des cartes serait aussi souhaitable .

interfaçage évolué : le langage qui sera utilisé pour développer et mettre au point les applications doit donner au programmeur d'application un haut niveau d'interfaçage .

évolutivité : ce système ne doit imposer aucunes limitations quant aux usages possibles de la carte à microprocesseur . Il doit pour cela s'adapter aux besoins du marché .

compacité : ce langage de programmation doit rester assez compact , ceci afin de permettre un téléchargement rapide dans le terminal-caisse et un stockage aisé des différentes applications dans les cartes applicatives ou dans le terminal-caisse .

téléchargement : le même langage doit pouvoir être utilisé afin d'écrire des applications locales sur un micro-ordinateur ou de les télécharger à partir d'un central . L'intérêt est ici de décharger le serveur de certaines tâches en décentralisant les traitements de l'application impliquant la carte et le terminal-caisse .

Pour réaliser ces buts , nous disposons et étudions l'architecture logicielle suivante :

- l'interpréteur , un logiciel qui décompose toute application SCIL en une suite d'ordres élémentaires et qui est situé dans le terminal-caisse .

- le SCIL , un langage standard permettant d'écrire des applications résidentes ou téléchargées . L'étude de ce langage nécessitera au préalable celle des mémoires de la macromachine et de la macromachine elle-même . Nous terminerons ce chapitre en examinant si les buts fixés ont été atteints .

- un ensemble de logiciels d'aide au développement d'application SCIL sur un micro-ordinateur de type P.C. (Assembleur , Editeur de liens, Simulateur). Ces derniers logiciels ne seront pas abordés dans ce mémoire .

8.3 L'INTERPRETEUR .

Voyons le principal problème qui se pose au programmeur puis sa solution .

8.3.1 LE PROBLEME PRINCIPAL.

Pour réaliser l'application monétaire , nous constatons que le processeur central du terminal-caisse est amené à devoir dialoguer avec une multitude de périphériques et de coupleurs de cartes (Cfr. 5.2.2 La configuration du système) .

Chaque périphérique ou coupleur possédant ses propres ordres , le développeur d'applications-carte doit connaître les interfaces énoncées ci-dessous :

- Interface carte porteur et applicative
- Interface clavier client
- Interface clavier commercant
- Interface écran client
- Interface écran commercant
- Interface mémoire des transactions
- Interface lecteur de piste magnétique
- Interface modem
- Interface lecteur de données biométriques
- Interface module de comparaison des données biométriques
- Interface imprimante
- Interface horodateur

Le travail du programmeur se complexifie si nous imaginons qu'une application quelconque doit être portable sur une gamme variée de terminaux-caisse . En effet , un terminal-caisse peut posséder :

- divers types de périphérique
- divers types de gestion de périphérique
- un nombre indéfini de périphériques d'un même type.

Que reste-t-il de cette portabilité d'une application si nous envisageons de la développer pour des terminaux-caisse possédant des processeurs différents .

Il est donc intéressant pour la personne concernée par le développement de l'application, de disposer d'un langage de programmation assez évolué. Ce langage lui masquant les détails des divers dialogues avec les périphériques et les cartes .

8.3.2 LA SOLUTION .

La solution est un jeu d'ordres standards destinés aux entrées/sorties cartes et un second jeu pour les entrées/sorties périphériques .

Dès lors , il suffit au programmeur de spécifier vers quel périphérique (n°) il envoie l'ordre standard d'entrée/sortie .

Mis à part ces deux jeux d'instructions évoluées gérant l'environnement du terminal-caisse , d'autres jeux d'instructions sont attachés à la gestion de la macromachine (interpréteur , chargement , unité arithmétique et logique , ..).

Le langage SCIL. se décompose en consignes et en instructions :

La figure ci-dessous illustre le routage des consignes et des instructions d'un programme applicatif :

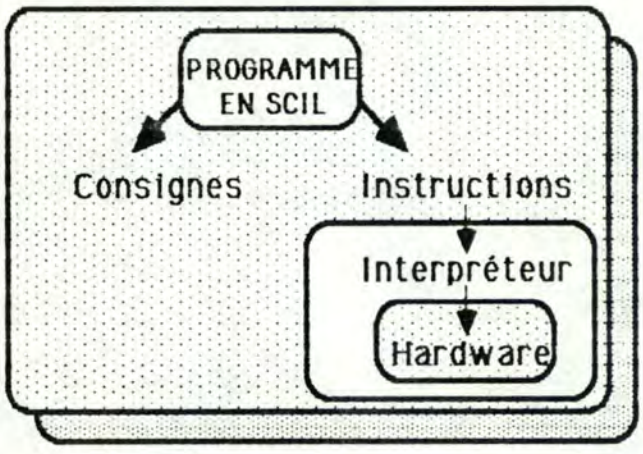


Fig. 8.1 Programme SCIL

- les consignes ne sont pas exécutées par le microprocesseur ,
- les instructions sont les opérations les plus élémentaires de l'interpréteur , elles sont toujours intégralement exécutées (interprétées) avant passage à l'instruction suivante . Elles sont relatives aux périphériques , aux cartes et à la macromachine .

La figure ci-dessous illustre le paramètrage des instructions par les consignes :

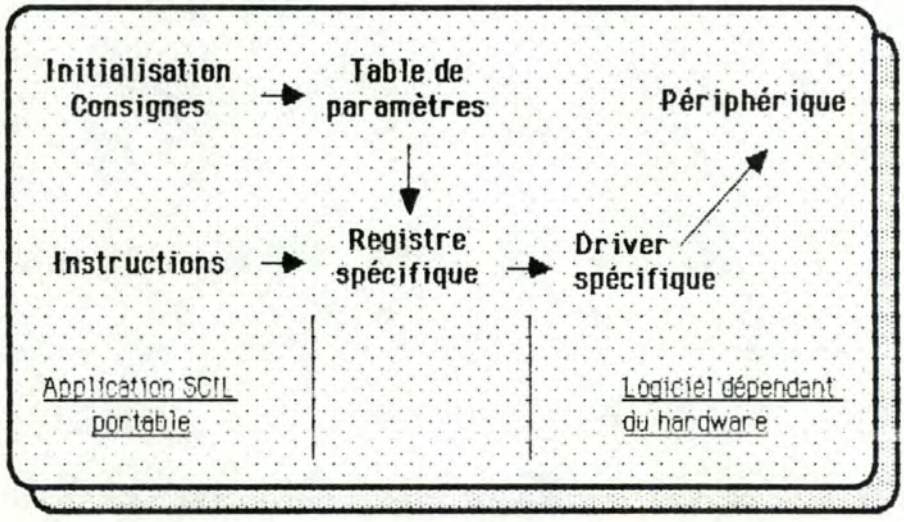


Fig. 8.2 Paramètrage des instructions par les consignes.

- les consignes assurent le dialogue entre l'application et la macromachine en initialisant le contexte (hardware et software) de la macromachine (Table des paramètres) .
- les instructions ont leur mode d'exécution défini par les consignes , ce qui permet un dialogue adéquat avec chaque type de driver .

Ainsi , tout programme applicatif dispose d'une indépendance vis-à-vis du hardware , des drivers des périphériques et des masques de carte , ce qui lui assure une portabilité .

Comme l'illustre le schéma suivant :

- l'interpréteur varie (B,C) selon le processeur-central (8035,8036) du terminal-caisse ,
- les consignes (CONS.) ne varient que selon les périphériques (A,B,...) ,
- les instructions (INS.) ne varient jamais .

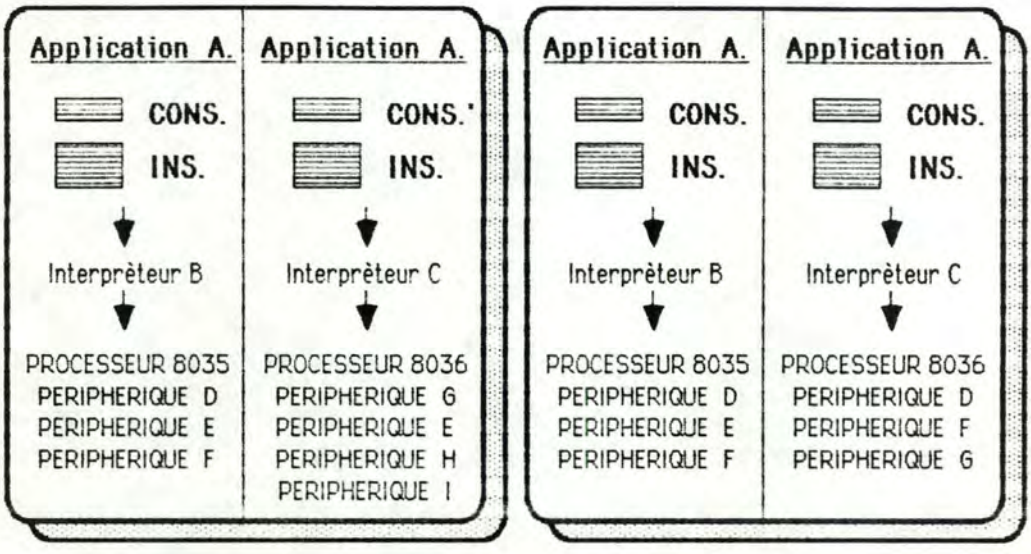


Fig. 8.3 La portabilité.

8.4 LES MEMOIRES DE LA MACROMACHINE .

Les objets manipulés par ce langage sont des informations situées à différents emplacements de la mémoire du terminal-caisse . Nous étudions pour cela :

- les zones de programmes (ou de données) ,
- les zones de registres ,
- la notion de classe d'une macromachine .

8.4.1 LA ZONE PROGRAMME.

8.4.1.1 LES TYPES ET LES SOUS-ZONES .

La macromachine dispose d'une zone programmes dont la taille est d'un mégaoctet . Elle est décomposée en 64 pages de 16 koctets chacunes . Ces pages sont réparties parmi les trois types de mémoires suivants :

- 20 pages pour la mémoire volatile (RAM)
- 20 pages pour la mémoire sauvegardée (RAM sauvegardée ou EEPROM)
- 20 pages pour la mémoire résidente (ROM ou PROM)

Une découpe plus fine de ces trois types de mémoires consiste à les décomposer chacune en trois sous-zones :

- Z.C. : cette sous-zone comprend des informations certifiées (programmes ou données) , c'est à dire des informations dont on est sûr de l'origine .
- Z.N.C : à l'inverse , cette sous-zone ne contient que des informations (programmes ou données) non certifiées ou non encore certifiées .
- Boite aux lettres : le troisième type de sous-zones contient des informations sensibles (programmes ou données) , d'où la présence d'une clé d'accès pour accéder à ces informations .

Le lecteur se reportera à la figure 8.4 pour une illustration de ce découpage de la zone programmes .

La gestion des boites aux lettres peut se résumer en trois actions .

A. La création .

Tout programme est autorisé à créer une boite aux lettres . Lors de cette création , il spécifie sa longueur et sa protection . Il fournit pour cette dernière la clé qui devra être présentée avant tout accès à cette boite aux lettres (en lecture ou en écriture) .

Une boite aux lettres peut aussi ne pas être protégée (cas d'une création sans présentation de clé) , l'accès à celle-ci est alors libre .

Le programme a aussi le choix entre créer cette zone dans la mémoire volatile ou sauvegardée . Dès lors , après une mise hors-tension du terminal-caisse , le système ne conservera , en permanence , les données et protections de cette boite aux lettres que si elle est en zone sauvegardée .

B. La fermeture .

La fermeture d'une boite aux lettres est réalisée lors d'une mise hors-tension du terminal-caisse , lors d'une fermeture explicite ou lors de la fin de session du programme l'ayant ouverte .

C. L'ouverture .

Afin de redonner l'accès libre à une boite aux lettres , il est au préalable nécessaire de respecter sa protection : c'est à dire de présenter la même clé d'accès que celle fournie lors de sa création (si une clé fut fournie , sinon l'accès est libre) .

8.4.1.2 LES NIVEAUX DE SECURITE DES PROGRAMMES .

Nous savons déja la nécessité de protéger certaines des informations (données ou programmes) contenues dans les diverses zones mémoires du terminal-caisse (Cfr. 7.1.3 La sécurité) .

Ces informations seront accessibles en lecture ou en écriture à certains programmes, alors qu'elles seront inaccessibles à d'autres . La macromachine distingue pour cela différents types de programmes et leur associe un niveau de sécurité déterminé . Ces niveaux leur allouent des droits d'accès différenciés à la mémoire .

Le niveau 0 : à ce niveau se situent les programmes les plus sûrs . Ce sont les programmes système tels que ceux de l'interpréteur ou des drivers .

Le niveau 1 : nous trouvons à ce niveau les programmes certifiés . Ce sont des programmes résidents au terminal-caisse (en mémoire ZC (ROM , PROM)) ou des programmes applicatifs téléchargés (en mémoire ZC (RAM)) et accompagnés d'une signature éléctronique .

Le niveau 2 : ce niveau identifie les programmes contenus dans une mémoire ZNC du terminal-caisse , mais qui sont non certifiés .

Le niveau 3 : ce sont les programmes du central , c'est-à-dire les instructions en provenance de l'ordinateur hôte .

Rem.

Le téléchargement d'informations (données ou programmes) à partir du central ou d'une carte applicative est autorisé vers des sous-zones de type non certifié (Z.N.C.) . Les programmes , dans ce cas , acquièrent le niveau 2 . Les données peuvent être ensuite transférées vers une sous-zone de type certifié (Z.C.) . Ce second transfert nécessite l'utilisation d'une routine du niveau 0 oul qui assure aussi l'authentification du central ou de la carte applicative.

On peut également envisager le téléchargement direct dans une zone ZC par paquets (de longueurs inférieures ou égales à 256 octets) . L'application exigeant de chaque paquet la détention d'une signature électronique identifiant l'émetteur (authentification de l'émetteur) .

8.4.1.3 LA PROTECTION DE LA MEMOIRE .

La figure ci-dessous illustre les droits d'accès à la mémoire en lecture ou en écriture des différents programmes selon leur niveau (en abcisses) . La mémoire est décomposée en sous-zones (ordonnée) .

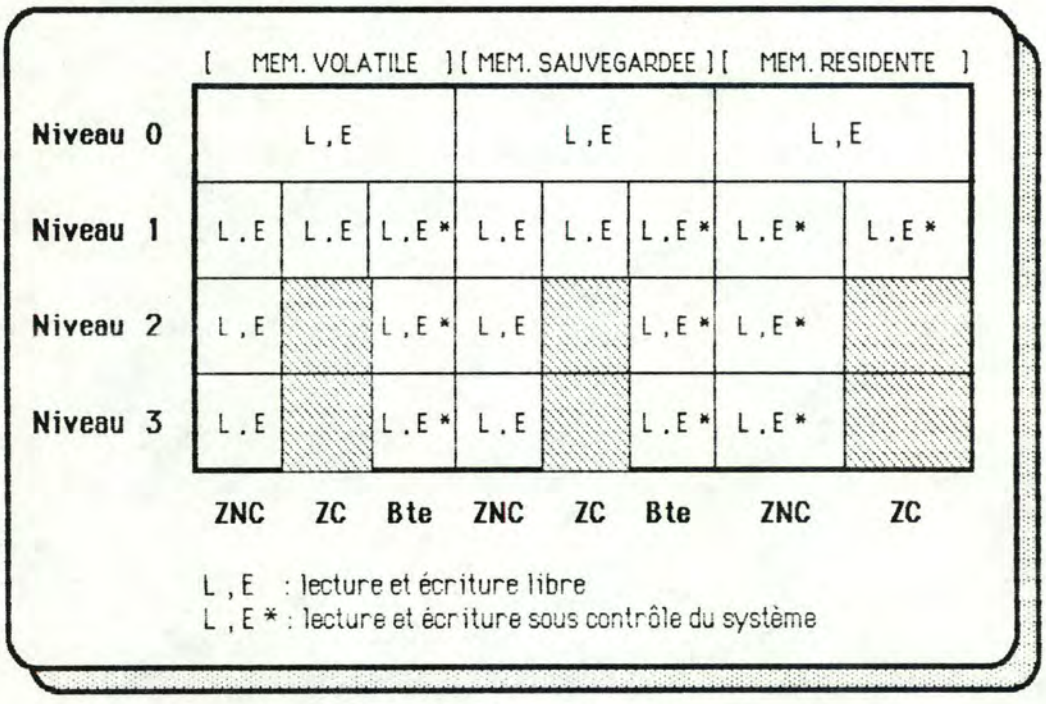


Fig. 8.4 Les accès des sous-zones selon le niveau.

- Les programmes qui ne sont pas au moins certifiés (niveaux 2 et 3) n'ont pas accès aux zones certifiées .
- Toutes les boites aux lettres peuvent avoir un accès protégé .
- Mis à part ces deux cas , toutes les autres zones sont accessibles librement par tous les programmes .

8.4.2 LA ZONE REGISTRE .

Mis à part la mémoire adressable d'un mégaoctet (zone programme) , la macromachine dispose également d'un espace registre (zone registre) .

Cet espace registre est décomposé en 16 bases de 256 registres chacune . Un registre est l'équivalent d'un octet . Un registre quelconque est donc adressé en spécifiant la base (0-15) à laquelle il appartient et son numéro (0-255) . Un bloc de registres est une collection de registres contigus qui est défini par sa longueur en nombre de registres et son adresse de début dans une base de registres .

La figure suivante illustre les différents types de bases .

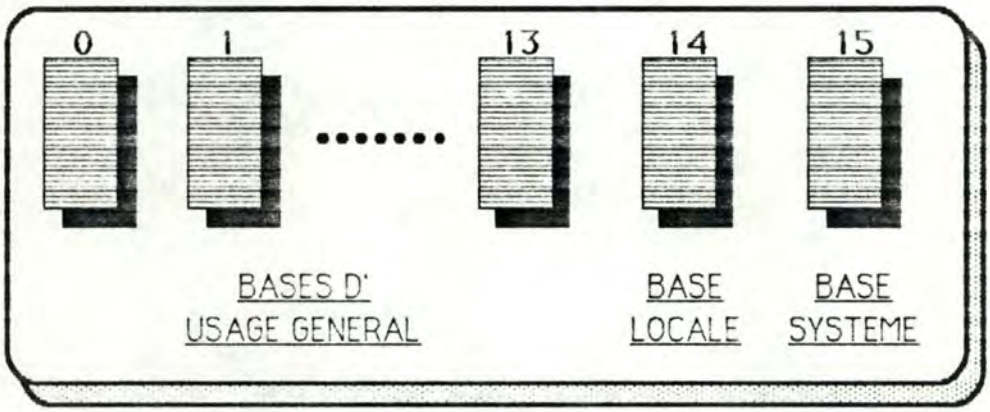


Fig. 8.5 Les bases de registres

- Les bases d'usage général sont utilisées comme bases de transfert (T) vers les divers périphériques et comme bases de travail (W) associées aux périphériques . On associe donc à un périphérique un couple W/T .
- La base locale est utilisée comme une base sécuritaire. En effet , toutes les informations introduites dans celle-ci ne peuvent être transférées que vers une carte à microprocesseur ou vers un module de chiffrement . De plus , certaines opérations sur son contenu sont interdites (Cfr. 8.7.3 La sécurité) .
- La base système est utilisée pour déterminer la configuration du terminal-caisse (allocation de la mémoire , table des ressources ...) .

La figure ci-dessous illustre une affectation possible des bases pour un ensemble de périphériques d'un terminal-caisse .

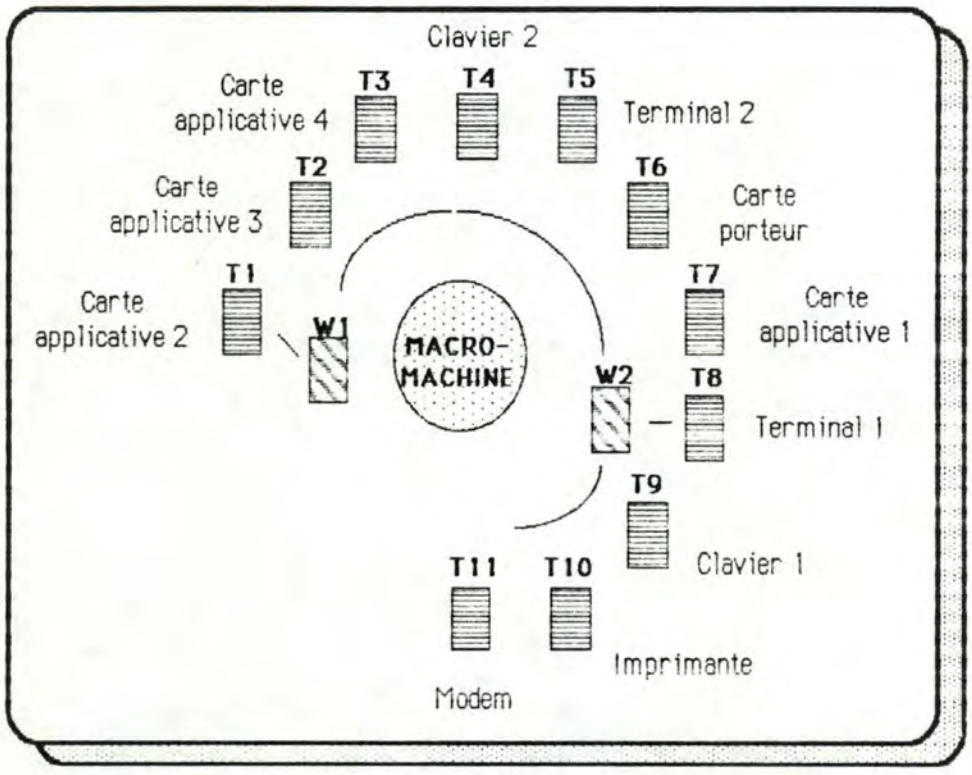


Fig. 8.6 Affectation des bases W et T

Avant chaque activation d'un périphérique , l'application définit dynamiquement quelle base elle choisit comme base de transfert (T) et de travail (W) pour dialoguer avec le périphérique concerné . La configuration peut aussi être monobase , dans ce cas , une base joue simultanément le rôle de base de travail et de transfert (base w = base T) .

8.4.3 LA CLASSE D'UNE MACROMACHINE.

Après avoir étudié la zone programmes et la zone registres , il nous faut parler de la notion de classe pour une macromachine .

Cette notion identifie la capacité du terminal-caisse . En effet , les divers terminaux-caisse ne pourront pas tous supporter cette taille mémoire et ce nombre de registres . Un terminal-caisse sera donc identifié par une classe . Une application SCIL développée pour un terminal-caisse d'une certaine classe (capacité) , sera portable sur tous les terminaux-caisse dont la classe lui est supérieure ou égale .

8.5 LA MACROMACHINE.

Pour une bonne compréhension de cette macromachine , ce point propose l'étude :

- de la gestion de l'environnement de la macromachine ,
- de la gestion des évènements en provenance de cet environnement ,
- de la dynamique des états de la macromachine ,
- de la gestion d'une application SCIL ,
- de la gestion de la transparence pour un flux de données interne à cet environnement ,
- des utilisations des bases à usage général ,
- des utilisations des bases système .

8.5.1 L'ENVIRONNEMENT .

Vu que la macromachine doit gérer un environnement assez hétéroclite , elle dispose d'une méthode standard pour paramètrer la multitude de drivers des périphériques . Elle admettre jusqu'à 256 périphériques et coupleurs de cartes . Ce point étudiera la méthode d'interfaçage dans ses grandes lignes .

Le réalisation de cet interfacage se décompose en plusieurs étapes .

Lors d'une première étape , tous les types possibles de périphériques (et cartes) ont été recensés dans une liste de codification des périphériques (terminal , imprimante , modem , chiffreur, horodateur , timer , carte , carte applicative) . De même , toutes les ressources pouvant exister dans un environnement SCIL ont été recensées dans une seconde liste (V 24 RS232 , VX ETTD, V24 ETCD , DIN , DES , RSA , GOC , horodateur , timer , ISAM) . Enfin , une troisième liste de tous les drivers possibles fut construite énumérant les différents types de gestion de l'interface physique des périphériques .

La seconde étape concerne une macromachine précise et son environnement Une table , énumérant les ressources dont la macromachine dispose , est construite . Cette table est stockée dans une mémoire résidente du terminal-caisse et ne peut être modifiée que par certaines routines (niveau 1) (cfr. 8.5.7. La base système).

C'est lors de la troisième étape , à chaque mise sous tension du terminal-caisse , que la macromachine construit la dernière table (table d'affectation des ressources) . Elle se base pour cela sur la table construite lors de la seconde étape . Cette nouvelle table est donc propre à la configuration sur chaque macromachine . Elle contient les informations de chaque périphérique pour réaliser l'interfaçage .

Chaque périphérique de l'environnement SCIL y est représenté par

- Di : le numéro identifiant le périphérique , Di = ti + ni :
  - ti : type de périphérique
  - ni : nième périphérique de type ti ,
- Dri : le numéro du driver utile pour la gestion du périphérique Di ,
- adr DDi : l'adresse du descripteur du périphérique Di (Cfr. 8.5.1.1)
- SI : le statut rendu par la macromachine après activation du périphérique Di (Cfr. 8.5.1.2).

8.5.1.1 LES DESCRIPTEURS (DDI) .

Le descripteur constitue le paramètrage de l'espace contexte du périphérique ou de la carte (d'ordre i de la macromachine) . Il est localisé dans la zone programmes .

A. Le descripteur de carte .

Dans le cas où le périphérique correspond à une carte , nous trouvons le descripteur de carte suivant (DDI) . Il comprend deux blocs :

- le bloc physique : il constitue l'espace contexte du coupleur associé à la carte Di ,
- le bloc surveillance : il constitue l'espace de contrôle de la carte Di .

Rem .

Le bloc de surveillance concerne la sécurité de la carte . La macromachine offre en effet la possibilité de contrôler toutes les mises sous tension de la carte Di et/ou toutes les émissions d'ordres entrant/sortant concernant la carte Di . Ce mode de surveillance est utilisé selon l'application , la surveillance peut être déclenchée à deux reprises :

- un branchement vers une routine certifiée (niv. 1) est produit lors de chaque mise sous tension de la carte .

- l'interpréteur surveille les ordres envoyés vers la carte . Si un ordre envoyé par l'application est à surveiller (c'est-à-dire s'il est présent dans le bloc de surveillance) un branchement vers l'adresse d'une routine se produit. Le bloc de surveillance contient donc une table des profils des ordres-carte à surveiller .

B. Le descripteur de périphérique .

Le descripteur de périphérique comprend principalement trois blocs :

- le bloc physique , il établit le contexte de la ressource associée au driver physique
- le bloc éditeur , il établit le contexte de l'éditeur du périphérique
- le bloc protocole , il établit le contexte du driver du périphérique

Rem .

Le bloc éditeur peut se décomposer en trois blocs de types divers :

- le bloc message , il contient l'ensemble des messages prédéfinis à émettre vers le périphérique Di ,
- le bloc fonction , il contient les fonctions caractéristiques du périphérique Di (fonction de validation , correction , annulation , ...)
- le bloc système , il contient diverses informations telles que les types de caractères autorisés à l'écran , l'écho masqué , time-out , ...

Le bloc protocole contient principalement le descripteur de transparence associé au driver . On y indique :

- les périphériques à utiliser pour le chiffrement et la compression ,
- les séquences de caractères échappant au chiffrement et à la compression ,
- d'autres conventions de dialogue .

8.5.1.2 LE STATUT DU PERIPHERIQUE :

Le statut du périphérique (SI) est mis à jour lors de la plupart des consignes d'initialisation , des entrées-sorties ou suite à un événement détecté concernant le périphérique Di .

Il nous renseigne si une erreur est détectée ou non , si le périphérique est connecté ou non , ... Mais les événements détectés varient aussi selon le type de périphérique :

Ex. : le caractère d'échappement détecté (terminal , modem),
la présence ou absence carte (carte) ,
la fin d'exécution (imprimante) .

8.5.2 LA TRANSPARENCE :

Lorsqu'une application non SCIL , distante ou non , désire utiliser un périphérique du terminal-caisse , celui-ci assure la transparence des données transmises d'un périphérique à un autre . Ce mode de transparence exige de la macromachine de ne pas activer son interpréteur lors de la réception des données (Ex . le modem émettant pour un terminal ou une imprimante) .

Les données reçues peuvent être retransmises vers plusieurs destinataires simultanément . Elles peuvent aussi être chiffrées , déchiffrées ou compressées avant d'être transmises .

La figure suivante illustre le parcours effectué par des données redirigées directement d'un périphérique à l'autre , ceci grâce au module de routage-cryptage .

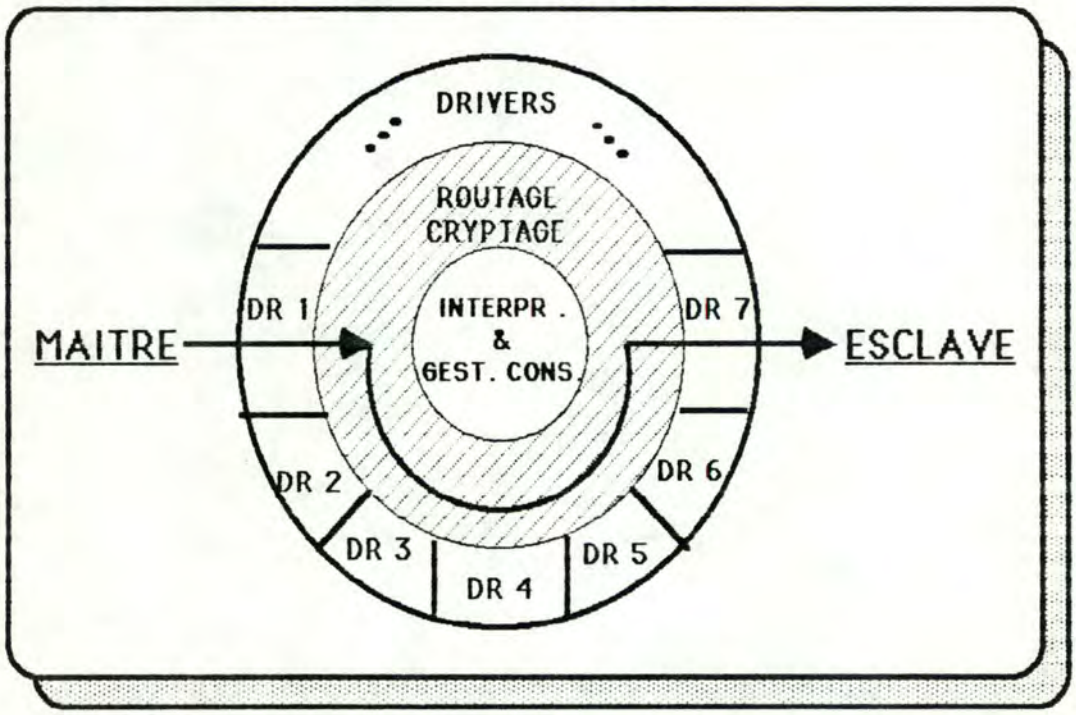


Fig. 8.7 La transparence

Le driver du périphérique maître , recevant les données de l'extérieur les envoie au module de routage-cryptage , via une interface appelée descripteur de transparence (Cfr. 8.5.1.1 Les descripteurs) . C'est dans ces interfaces que l'on réalise le paramétrage de la transparence (chiffrement , compression , ...) . Les données sont ensuites envoyées au(x) driver(s) destinataire(s) (périphérique esclave) via leur interface . Les deux drivers communiquent donc directement , sans passer par l'interpréteur .

8.5.3 L'ETAT DE LA MACROMACHINE .

La macromachine ainsi que les périphériques peuvent se trouver dans trois états :

- l'état de session maître ,
- l'état de session esclave ,
- l'état hors session .

Les prérogatives d'un périphérique se trouvant en état de session maître sont supérieures à celles des autres périphériques .

La dynamique des états de la macromachine est fonction :

- d'une part des consignes reçues et acceptées par elle . Ce sont les consignes (cfr. Annexe II) :

  - de mise en mode de la macromachine (CM)
  - de fin de mode de la macromachine (FM)
  - de réinitialisation de la macromachine (Reset)
  - de déconnexion d'un périphérique (CNX of)
  - de mise sous tension de la macromachine (BOOT)

- d'autre part de l'origine de ces consignes . L'application (ou routine) émettrice peut être :

  - interne à la macromachine ,
  - un périphérique en session maître ,
  - un périphérique en session esclave ,
  - un périphérique externe (non concerné par la session courante) .

Le diagramme d'état suivant illustre cette dynamique des états de la macromachine utilisée pour une application .

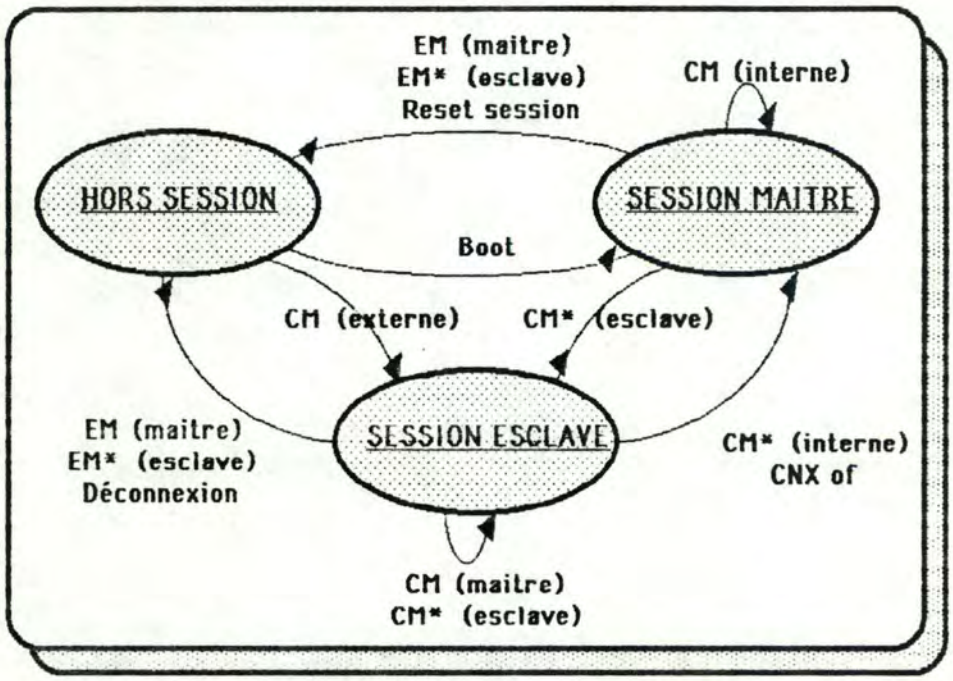


Fig. 8.8 Les états de la macromachine

- CM (interne) : instruction équivalente à une consigne interne de mise en mode
- EM (interne) : instruction équivalente à une consigne interne de fin de mode
- CM (maître) : consigne de mise en mode en provenance d'un périphérique maître
- EM (maître) : consigne de fin de mode en provenance d'un périphérique maître
- CM (esclave) : consigne de mise de mode en provenance d'un périphérique esclave
- EM (esclave) : consigne de fin de mode en provenance d' un périphérique esclave
- CM (externe) : consigne de mise en mode en provenance d'un périphérique externe
- EM (externe) : consigne de fin de mode en provenance d'un périphérique externe
- * : après acceptation de la consigne par la macromachine
- CNX of : instruction de déconnexion venant de l'application esclave

8.5.4 LES APPLICATIONS SCIL .

Les applications SCIL débutent par une ouverture de session et se terminent par une fermeture de session . L'ouverture de session est provoquée explicitement par une consigne de mise en mode (CM) . La fermeture de session est provoquée explicitement par une consigne de fin de mode (FM) ou par une réinitialisation (Reset) .

Lors d'une session , l'application composée de consignes et d'instructions peut passer indifféremment par une des trois phases suivantes :

- phase où le gestionnaire de consignes est actif
- phase où le gestionnaire d'instructions (l'interpréteur) est actif
- phase intermédiaire

Cette dernière phase est interrompue lorsqu'une consigne ou une instruction arrive en provenance d'un périphérique sous surveillance . Selon le type de données reçues , une des deux autres phases est activée .

En général , une application se déroule entièrement en session maître ou esclave . Une application en session esclave peut se terminer en session maître si elle déconnecte le périphérique abritant l'application maître et si cette déconnexion est acceptée par le terminal-caisse (CNX of) .
L'enchainement d'application est possible grâce à un ordre de lancement d'application et une gestion des reprises extérieures .

8.5.5 LES EVENEMENTS .

Une application peut contrôler les événements survenants sur les périphériques : la macromachine active , si nécessaire , son interpréteur pour traiter l'événement .
Suivant le type de périphérique , les évènements recensés sont :

- une connexion de périphérique
- une déconnexion de périphérique
- une détection d'un caractère d'échappement
- une insertion de carte
- un arrachement de carte
- une fin d'exécution de périphérique (imprimante) .

L'application peut se trouver lors de cette détection , dans une quelconque des trois phase énumérées au point 8.5.4 .

Le périphérique concerné (Di) et le type d'évènement survenu (Si) sur ce premier sont indiqués dans les deux premiers registres (R0 et R1) de la base de travail courante . Le registre R1 contient les mêmes informations que l'indicateur SI (Cfr. 8.5.5 Descripteur de périphérique) mais est associé au périphérique courant .

Plusieurs évènements simultanés sur divers périphériques sont traités dans l'ordre de priorité défini par les numéros Di identifiant chaque périphérique .

8.5.6 LES BASES D'USAGE GENERAL .

Nous allons étudier dans ce paragraphe l'organisation de l'espace registres lors des opérations d'entrées- sorties carte et périphérique .

Nous savons déjà que lors d'un dialogue avec un périphérique quelconque, l'application doit déterminer la base de registres qu'elle utilisera pour ses transferts et celle pour ses données de travail (Cfr. 8.4.2 La zone registres) .

L'application SCIL dialogue avec un périphérique et une carte , ceci via une base de travail et de transfert . Lorsqu'elle change de périphérique ou de carte elle change ou non de bases de travail et de transfert .

Afin de simplifier l'étude , nous posons l'hypothèse suivante : la base de travail est différente de celle de transfert . Nous ne sommes donc pas dans une configuration monobase .

Voyons successivement le contexte d'un dialogue avec une carte à microprocesseur puis celui avec tout autre périphérique .

8.5.6.1 LE DIALOGUE CARTE .

Une opération d'entrée/sortie carte correspond à une instruction simple ou basée sur l'algorithme Télépass . Deux types d'informations sont alors échangées avec la carte à microprocesseur :

- les paramètres de l'instruction envoyée

- les données - entrantes (à soumettre à la carte)
  - sortantes (en provenance de la carte)

La figure suivante les illustre , ainsi que les endroits dans la base de travail et de transfert où sont localisés ces deux types d'informations .

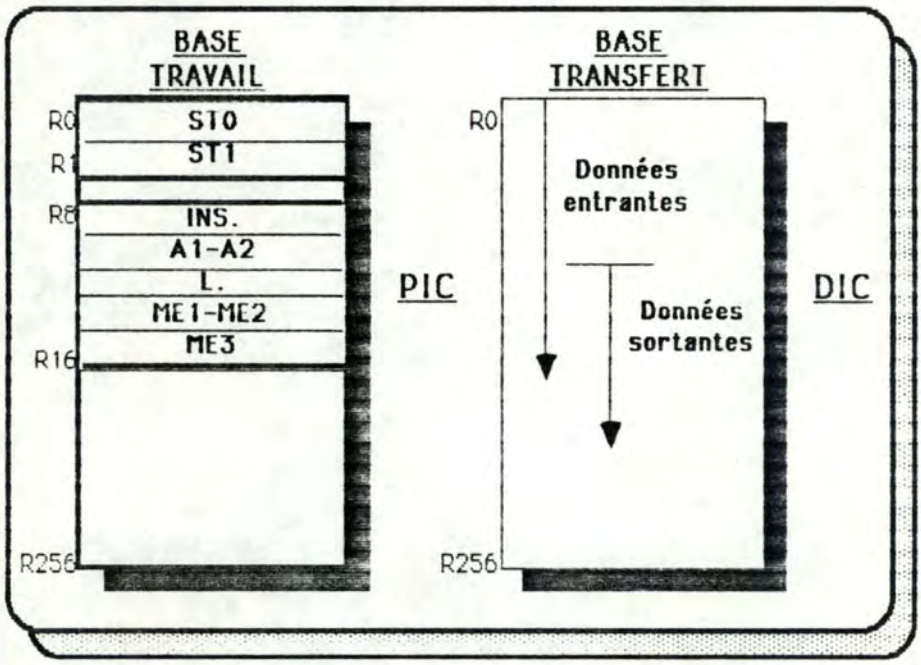


Fig. 8.9 Les Bases de travail et de transfert

- RO est l'identifiant du périphérique courant (un carte ou un périphérique)
- R1 est le statut de cette carte ou périphérique
- PIC (**p**aramètres d'**i**nterface avec la **c**arte) . Ils comprennent :

  - INS : l'instruction carte à activer (Lect. , Ecr. , ...)
  - A1-A2 : le paramètre adresse de cette instruction ,
  - L : le paramètre longueur de cette instruction ,
  - ME1-ME2 : les mots d'état de la carte ,
  - ME3 : le mot d'état du coupleur de la carte .

- DIC (**d**onnées entrantes ou sortantes d'**i**nterface avec la **c**arte) Elles sont échangées avec la carte . Lors d'une instruction algorithmique , l'identifiant de la clé à utiliser se trouve dans le premier mot des données entrantes . Viennent ensuite les données entrantes à soumettre à la carte , pour la réalisation de l'ordre carte .

8.5.6.2 LE DIALOGUE PERIPHERIQUE .

Dans le cas d'un dialogue avec un autre type de périphérique , nous ne rencontrons plus les paramètres d'interface avec la carte (PIC) . Il subsiste tout de même les données envoyées vers ou reçues de la carte (DIC).

En bref : le dialogue vers une carte ou un périphérique doit de toute façon se décomposer en deux étapes :

1. charger les registres appropriées avec les paramètres de l'ordre (PIC et/ou DIC) ,
2. activer l'ordre de sortie vers la carte ou le périphérique (Cfr. Annexe II Les instructions d'entrées-sorties carte ou périphérique : INC , OUTC , IN ...) .

Lors de la réponse , la procédure est inversée , l'ordre est celui d'entrée et il suffit de lire les données reçues dans les registres DIC .

8.5.7 LA BASE SYSTEME .

La base système combinée avec les descripteurs de périphérique ou de carte définissent l'interface paramètre entre la macromachine , l'interpréteur et les drivers . Elle est divisée en trois espaces de contenus et de protection différents .

La figure ci-dessous illustre ces trois espaces .

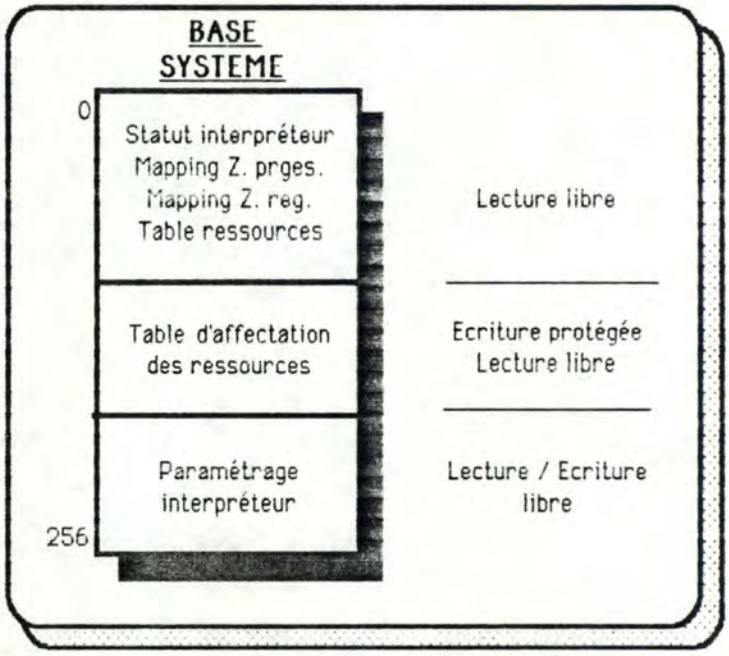


Fig. 8.10 La base système

La première partie contient :

- le statut de l'interpréteur
- la classe de la macromachine (taille de la mémoire , nombre de bases de registres , ...)
- l'environnement de la macromachine (Cfr. 8.5.1 2° Etape , la table des ressources)

La seconde partie contient le paramétrage de cet environnement (Cfr. 8.5.1 3° Etape , la table d'affectation des ressources AF. B.) .

La troisième partie contient des paramètres de l'interpréteur (coupleur carte ou périphérique courant , chiffrement courant , bases de travail et de transfert courantes ...) .

8.6 LE LANGAGE .

Pour rappel les consignes établissent le contexte dans lequel les instructions seront exécutées .

Nous donnons ci-dessous une classification de ces consignes et instructions . Le lecteur se repportera à la seconde annexe pour plus de renseignements (Cfr. annexe 2 : Liste des instructions et des consignes du langage SCIL) .

8.6.1 LES INSTRUCTIONS .

1) Les instructions de contrôle : elles concernent le déroulement de l'interpréteur , du gestionnaire de consignes , la gestion globale des bases de registre , des évènements et de la mémoire programmes (boite aux lettres ...) .

2) Les instructions de chargement registres : elles concernent l'initialisation et le chargement des registres des différentes bases .

3) Les instructions de transfert de données : elles concernent les transferts d'informations internes à la zone registres , à la zone programmes et entre ces deux zones .

4) Les instructions arithmétiques et logiques : elles concernent les instructions arithmétiques et logiques classiques , exécutables par le processeur central .

5) Les utilitaires binaires : ce sont les opérations binaires (SHIFT , ROTATE) et les conversions arithmétiques sur les bases de registres .

6) Les instructions de branchement : ce sont les appels de sous-routine , les débranchements simples et les débranchements conditionnels .

7) Les instructions d'entrées-sorties carte : ce sont les mises sous et hors tension de la carte , l'envoi de toutes les instructions disponibles dans le jeu d'instructions d'une carte et la réception des données .

8) Les instructions d'entrées-sorties macromachine : ce sont l'envoi d'ordres vers les divers périphériques et la réception des données .

8.6.2 LES CONSIGNES

1) Les consignes de mise en mode : elles placent la macromachine en mode SCIL , déclenchent le mode Debbug de l'interpréteur .

2) Les consignes de chargement : elles concernent le téléchargement chiffré ou non de programmes ou de données dans la zone programme , l'envoi de données vers le périphérique maître et le téléchargement de données dans les bases de registres .

3) Les consignes de gestion des périphériques : elles concernent la mise à jour des descripteurs de périphérique ou de carte et l'initialisation de ces derniers .

8.7 EVALUATION

Nous allons , pour cette évaluation , passer en revue les buts fixés lors du point 8.2 (la compacité , le téléchargement , la sécurité , la portabilité et l'interfaçage évolué , l'évolutivité) , puis nous terminerons par les performances du système .

Il est important de spécifier que cette évaluation se base sur l'expérience acquise lors du développement d'une application . C'est-à-dire , l'application de paiement électronique (vue au chapitre 5) , dialoguant avec des cartes possédant le masque M.A. (vu au chapitre 6) et pour un terminal-caisse abritant la macromachine et le langage SCIL. (vu au chapitre 8) . Cette évaluation est donc légèrement biaisée , car elle se situe dans un contexte restrictif d'utilisation des divers composants .

8.7.1 LA COMPACITE

Mis à part les tables d'états qui pourraient être modifiées afin de les réduire, le langage SCIL possède des instructions très compactes , lorsque l'on considère le paramètrage qui les accompagne .

La réalisation d'une semblable application de paiement de contact , nous révèle qu'elle peut être contenue dans un programme SCIL de 4 Koctets au maximum .

Il est important de noter que la possibilité de fournir à certaines instructions des paramètres par adresse nous a permis de réaliser des modules de bas niveaux réutilisables dans divers contextes .

Cette compacité est aussi rendue possible par la présence des descripteurs des périphériques qui déchargent l'application d'une gestion fastidieuse de l'environnement .

On peut reprocher aux instructions de ce langage d'être parfois trop condensées , au point que les programmes SCIL seront difficilement maintenables .

8.7.2 LE TELECHARGEMENT

Le chargement de données ou de programmes à partir de la carte applicative , d'un central ou de tout autre périphérique est rendu possible grâce à des consignes de chargement dans la zone programmes ou registres (DLP , DLR) .

Un programme peut donc être développé à l'extérieur du terminal-caisse et être téléchargé , via le réseau ou la carte applicative . Cette seconde solution permet donc à un terminal-caisse de fonctionner en étant complètement séparé du central (hors-ligne) .

La compacité des applications développées en SCIL est avantageuse pour le téléchargement .

8.7.3 LA SECURITE

La macromachine apporte un dernier niveau de sécurité au système nécessaire à une telle application . Cette sécurité concerne les informations détenues par la macromachine , les programmes internes à lcette dernière et les instructions émises vers les cartes à microprocesseur .

1) Sécurité des informations détenues par la macromachine .

-› Les informations présentes dans la zone programmes sont protégées de diverses manières :

- grâce au niveau de sécurité : les effractions dues à des logiciels dont l'origine est douteuse (logiciels non certifiés) sont impossibles . En effet , les accès (aux zones programmes) accordés à une routine varient selon son niveau .

- grâce à l'instruction P.B. , qui rend un bloc de la zone programmes inaccessible en lecture ou en écriture par un niveau inférieur à 0 (1-2-3) . Le bloc reste seulement exécutable : ce qui constitue une protection contre les effractions logicielles .

- grâce à la présence de boites aux lettres , qui peuvent être sécurisées par leur clé d'accès et par les accès . Les boites aux lettres permettent aussi des communications inter-sessions très sécurisées . En effet , elles permettent le transfert d'informations d'une application à une autre (ex. stockage de transactions , de clés , de programmes) .

-> Seuls les registres de la base locale sont protégés . Cette base de registres est en effet destinée au stockage des informations confidentielles saisies au clavier , dans la carte du client , ...

Aucune donnée ne peut être transmise de cette base vers l'extérieur . De plus , toutes les instructions de lectures ou assimilées , de tests , d'opérations binaires , de sauts conditionnels sont interdites sur son contenu . Ceci afin d'éviter à un programme de deviner le contenu de ces registres .

2) Sécurité de routines internes

Certaines routines de sécurité sont prévues telles que la gestion des échos masqués d'un écran , utilisée lors de la saisie de la clé porteur .

De même , lors de cette saisie , une lampe est allumée pour assurer l'utilisateur que les données qu'il va introduire au clavier seront stockées en sécurité . Cette gestion de la lampe externe n'est accessible que par un programme de niveau 0 ou 1 .

La routine qui transfère un programme de la zone non certifiée (ZNC) vers la zone certifiée (ZC) doit être du niveau1 ou 0 .

3) Sécurité de la carte

Certains ordres cartes doivent être surveillés de près par la macromachine . Elle dispose pour cela d'une table lui indiquant tous les ordres à surveiller dans les dialogues avec une carte (bloc de surveillance) (cfr. 8.5.1.1 Les descripteurs).

8.7.4 PORTABILITE

La portabilité des applications dans différents environnements concerne , à la fois , le processeur central , les masques de carte et les divers périphériques :

- la portabilité vis à vis du processeur central du terminal-caisse est laissée aux bons soins de l'interpréteur .

- la portabilité vis à vis des masques de cartes est réalisée , mais cela ne signifie pas qu'un programme développé pour un masque précis sera totalement compatible avec un autre masque . Par exemple , un ordre de recherche sur argument ne sera compréhensible que pour la carte possèdant le masque MA. .
  L'avantage de ce langage pour tous ces masques est une formalisation commune de l'envoi d'un ordre-carte (INC , OUTC) (Cfr. 8.8.6.1 Le dialogue carte)

- La portabilité vis à vis des autres périphériques a été rendue possible grâce à la présence des descripteurs de périphérique ou de carte . Ils sont assez complexes à formaliser vu qu'ils représentent l'interface commune à l'ensemble de tous les types de périphériques (imprimante , terminal , ...) ou de toutes les cartes . Un ordre à destination d'un type de terminal pourra être expédié vers un autre type de terminal à condition que l'interface de chacun soit réalisé .

8.7.5 INTERFACAGE EVOLUE

Mis à part les instructions de gestion de la macromachine (contrôle , transfert , arythmétiques et logiques , utilitaires binaires , branchement) qui sont des macro-instructions classiques , les ordres à destination de l'extérieur (IN, INC , OUT , OUTC) sont d'un niveau assez élevé (ceci grâce à la présence de descripteurs) . Tous ces ordres sont très souples grâce à leur paramètrage .

Le programmeur évite de devoir gérer les fonctions de messagerie , il peut créer et stocker des messages prédéfinis dans une liste (Cfr. 8.5.1.1 Les descripteurs : bloc message , bloc fonction , bloc système) .

Il est utile de signaler le rôle important que jouent les tables d'états lors des dialogues avec la carte . Les tables d'états réalisent des tests successifs sur les mots états renvoyés par la carte (ME1 et ME2) et par son coupleur (ME3) . Ces mots nous renseignant sur les erreurs de types applicatif et matériel , il est regrettable que les tables d'états ne distinguent pas ces deux types d'erreurs (cfr. Annexe ME1 , ME2 , ME3) . Les tests réalisés par une table d'états sur les mots ME1 - ME2 - ME3 peuvent être déclenchés :

- explicitement , grâce à l'ordre BRST (cfr. Annexe 2)
- implicitement

Le principe implique que lors de tout ordre à destination de la carte , une table d'état soit référencée . La macromachine ne se contente donc pas d'envoyer , par exemple , un simple ordre de lecture , mais teste au retour les mots d'états et effectue les débranchements nécessaires . L'envoi d'un ordre-carte est donc performant , mais assez complexe pour un débutant .

Deux critiques leur sont adressables :

- une souplesse supplémentaire dans la formulation des conditions à évaluer dans chaque table est souhaitable .
- la compacité des tables peut être améliorée , l'application développée a nécessité la création de quatres tables pour la carte porteur (soit 295 octets). Elles ne peuvent donc pas se trouver en permanence et simultanément dans la base de travail associée à la carte porteur .

8.7.6 EVOLUTIVITE

Le système de la macromachine et son langage sont rendus libres de toutes contraintes .

La gamme variée de mémoires (programme ou registre) , de protections et le langage permettent à toutes applications d'être développées sur une telle machine .

Lors d'une session SCIL , la macromachine peut fonctionner dans le mode interpréteur , gestionnaire de consignes ou transparent . De plus , elle est libre d'être utilisée hors-session , comme tout autre système informatique .

8.7.7 PERFORMANCE

La liste suivante énumère des organisations ou des instructions dont l'usage fut décisif pour le développement de l'application de paiement de contact .

-> Les bases : l'utilisation courante et aisée de celles-ci dans un un environnement d'une dizaine de périphériques fut la suivante :

- une base de registre utilisée , en permanence , comme base locale ,
- une autre , en permanence , comme base de travail ,
- six autres , en alternance , comme base de transfert .

Les bases de transfert n'ont jamais utilisé plus de 32 de leurs registres sur les 256 qu'elles contiennent . La base locale n'en utilisait que 10 au maximum, par contre , la base de travail fut débordée .

L'allocation dynamique de bases de transfert et de travail au périphérique courant , autorise le programme à dialoguer " simultanément " avec plusieurs cartes . Il lui suffit avant l'émission d'un message vers une autre carte : de spécifier la carte avec laquelle il veut dialoguer et de modifier la base de transfert .

-> Une instruction (FTYPE) permet une recherche sur argument assez performante dans la zone programmes . Elle s'adapte très bien à la recherche dans une structure arborescente d'un même type que celle des descripteurs de périphérique : (descripteur DI (bloc physique (- , -) ; bloc éditeur (bloc fonction , bloc message , bloc système) , bloc protocole (bloc transparence) .

-> L'instruction (EVENT) permet à la macromachine de gérer en multitâche des drivers de type lent (ex. imprimante) . Il lui suffit d'envoyer son ordre et ses données et de mettre le périphérique sous surveillance en attente de l'événement fin d'exécution . Entre-temps elle est libre de se consacrer à d'autres tâches .

-> Les routines résidentes : le terminal-caisse peut stocker toutes sortes de routines dans sa mémoire résidente . Ces routines peuvent être rédigées en assembleur ou en langage SCIL .

L'instruction EXEC autorisera le lancement à partir de programmes SCIL des routines assembleurs , les autres nécessitant l'utilisation de l'interpréteur .

Ces routines peuvent constituer des modules de traitements associés à diverses gestions :

- gestion des impressions ,
- gestion de la mémoire des transactions ,
- gestion du clavier et de l'écran ,
- gestion des autotests ,
- gestion des authentifications de terminal , du porteur ,
- gestion des changements de clé porteur .

La présence de ces routines , résidentes au terminal-caisse entrainerait une diminution de la taille des applications à télécharger .

Conclusions du mémoire

Voici venu le moment de faire le bilan des résultats obtenus concernant la carte à microprocesseur .

La synergie de son électronique , de l'informatique et du chiffrement réalise une très haute protection , dont on peut aujourd'hui entourer les informations . Ce type de carte semble être une parade adéquate à la contrefaçon , la manipulation et la simulation .

Mais considérant les systèmes actuels , l'amélioration de la sécurité ne justifie pas assez l'introduction , dans notre vie , de la carte à microprocesseur. Elle doit donc se tourner vers les applications utilisant son aspect polyvalent et multi-usages .

Le choix de mon mémoire m'a sensibilisé aux problèmes d'organisation et de sécurité qu'occasionne le développement d'une application de paiement électronique ; application dont pratiquement tous les risques ont été supprimés grâce a une intense coopération de la carte porteur , applicative et du langage utilisé pour la développer (SCIL) .

Après que la carte à microprocesseur ait déplacé les procédures de sécurité à son niveau (et non plus à celui du lecteur-encodeur) le langage SCIL , la carte applicative et le terminal-caisse sont parvenus à déplacer celles du central vers le site local (Ex. transactions hors-ligne) . Ce qui permet de réaliser des gains au niveau des coûts de télécommunication .

Un des points non abordé dans ce mémoire et qu'il serait intéressant d'approfondir est l'aspect juridique dû aux capacités mémoire de la carte à microprocesseur .

L'avenir de la carte à microprocesseur est déjà ébranlé par l'apparition de cartes à hautes performances , plus rapides où le dialogue avec le lecteur-encodeur se ferait grâce à un émetteur logé dans les cartes .

La carte à microprocesseur , bien que déjà agée (12 ans) , est une technologie qui a précédé la demande . D'où , même si elle devenait plus adéquate et possédait son cadre juridique , notre société et l'économie devraient encore l'accepter .

Bibliographie

DOCUMENTATION BULL CP8

Cours de formation

- Introduction à la carte CP8 .
- La carte M4 et B1 .
- Le SCIL et la macromachine .

Documentation technique

- Bull CP8 , une technologie nouvelle au service des nouvelles technologies .
- La carte CP8 .
- La carte CP8 , badge de haute sécurité .
- Le masque B2 , fonctionnalités .
- Le masque MA .
- La carte porte-clés vidéotex .
- Spécifications d'utilisation de la carte Bi-services GIE/PUBLIPHONE M4
- Processeur de sécurité associé .
- Card acceptor device application module .
- Preliminary specifications of the CP8 card .
- Guide utilisateur de la carte CP8 masque 4 .
- Lecteur de cartes à microcircuit pour Minitel , description de l'interpréteur.
- Spécifications d'interface , lecteur de cartes à microcircuit pour Minitel .
- Support de cours LECAM .
- LECAM/SCIL . Les objectifs .
- Smart card interpreted language .
- LECAM/SCIL . La concurrence .
- Interpréteur Lecam : extensions minimales .
- La macromachine .
- Le langage SCIL .

DIVERS

- Micro card technologies .
- Electronic signals and exchange protocols (Draft proposal 150 7816/3).
- Mastercard international I.C. card system .
- La carte à mémoire bancaire : les spécifications et les normes . (groupement carte à mémoire)

Table des matières

CHAPITRE 3 : DESCRIPTION TECHNIQUE

3.1. ARCHITECTURE INTERNE DE LA PUCE

3.2. DESCRIPTION D'UN MASQUE

3.3. LES PROGRAMMES D'APPLICATION

CHAPITRE 5 : LE PAIEMENT ELECTRONIQUE

5.1. LA CARTE ET LE MONDE BANCAIRE

5.2. LE PAIEMENT DE CONTACT

CHAPITRE 6 : L'EVOLUTION DES CARTES

6.I. LA DESCRIPTION DU MASQUE M.A.

6.2. LES AVANTAGES DU MASQUE M.A.

6.3. CONCLUSIONS

CHAPITRE 7 : LE MATERIEL

CHAPITRE 8 : LE LANGAGE

ANNEXES

1. LA GESTION DES ECHANGES CARTE LECTEUR-ENCODEUR

I. LES NORMES ISO : 7816/3
II. LES NORMES ISO : 7816/3 ADDENDUM 1

2. LES CONSIGNES ET LES INSTRUCTIONS

I. Les instructions de contrôle
II. Les instructions de chargement de registres
III. Les instructions de transfert de données
IV. Les instructions arythmétiques et logiques
V. Les utilitaires binaires
VI. Les instructions de branchement
VII. Les instructions d'entrées-sorties cartes
VIII. Les instructions d'entrées-sorties macromachine
IX. Les consignes

ANNEXE I :
LA GESTION DES ECHANGES
CARTE LECTEUR-ENCODEUR

I. Les normes ISO : 7816/3
II. Les normes ISO : 7816/3 ADDENDUM 1

ISO/TC 97/SC 17/WG 4 N 260

DRAFT PROPOSAL ISO/DP 7816/3	
date 1986-10-09	ISO/TC 97/SC 17
superseded document 97/17/4 N 244	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/TC 97/SC 17

"IDENTIFICATION AND CREDIT CARDS"

Secretariat B.S.I.

Circulated to P- and O-members, technical committees and international organizations in liaison for :

- discussion at
- comments by
- voting by (P-members only)

Title IDENTIFICATION CARDS - INTEGRATED CIRCUIT(S) CARD WITH CONTACTS

PART 3 : ELECTRONIC SIGNALS AND EXCHANGE PROTOCOLS.

Introductory note

IDENTIFICATION CARDS - INTEGRATED CIRCUIT(S) CARD WITH CONTACTS-
PART 3 : ELECTRONIC SIGNALS AND EXCHANGE PROTOCOLS.

0 - INTRODUCTION

This International Standard is one of a series of standards describing the parameters for integrated circuit cards with contacts and the use of such cards for international interchange. These IC cards are identification cards intended for transactions negotiated between the outside and the integrated circuit in the card. As a result of a transaction, the card delivers information (computation results, stored data), and/or modifies its content (data storage, event memorisation).

1 - SCOPE AND FIELD OF APPLICATION

This part 3 of ISO 7816 specifies the power and signal structures, and data and command structures for a preliminary dialogue between an integrated circuit(s) card and an interface device such as a terminal.

It will cover power, signal rates and voltage levels, parity conventions, operation procedures, transmission mechanisms, provisions for naming applications, additional aspects of data interchange format standardization and the organisation of communication with the IC card. It does not cover information and instruction content, such as identification of issuers and users, services and limits, security features, journaling, and those instruction definitions and instruction processing, which are defined in existing standards and in additional standards to be developed.

Illustration of the specifications of this standard is shown in annex 1 which does not form part of this standard.

2 - REFERENCES

ISO/ 1177- " Information processing- Character structure for start/stop and synchronous transmission".

ISO/7810 - "Identification cards - Physical characteristics"

ISO/DIS 7816/1 - "Identification Cards - Integrated circuit(s) card with contacts - Part 1 : Physical characteristics"

ISO/DIS 7816/2.2 - "Identification Cards - Integrated circuit(s) card with contacts - Part 2 : Dimensions and location of the contacts".

3 - DEFINITIONS

The term identification card is defined in ISO/DIS 7810.

Interface device : a terminal, communication device or machine to which the integrated circuit card is electrically connected during operation.

4 - ELECTRICAL CHARACTERISTICS OF THE CONTACTS

4.1 Electrical Functions

Contacts assignments are specified in part 2 of ISO 7816, supporting at least the following electrical circuits:

I/O : Circuit by which serial data is input to or output from the integrated circuit inside the card.

VPP : The programming supply voltage Vpp (optional use by the card).

GND : Zero voltage as a reference voltage (0 V).

CLK : Clocking or timing signal (optional use by the card).

RST : Circuit either used by itself (reset signal) or in combination with an additional control circuit to define specific function modes (optional use by the card, provided external Vcc supplies the reset).

VCC : The circuit supply voltage (optional use by the card).

Note : The use of two remaining contacts should be defined in the appropriate application standards and submitted to ISO TC 97 SC 17 for approval.

4.2 Voltage and Current Values

All measurements are defined with respect to contact GND and in a temperature range of 0°C to 50°C.

All currents flowing into the card are considered positive.

All timings shall be measured relative to the appropriate treshold levels as defined in 4.2.2 to 4.2.5.

4.2.1 Abbreviations used

V_{IH} stands for high level input voltage

V_{IL} stands for low level input voltage

I_{CC} stands for supply current

I_{IH} stands for high level input current

I_{IL} stands for low level input current

V_{OH} stands for high level output voltage

V_{OL} stands for low level output voltage

I_{OH} stands for high level output current

I_{OL} stands for low level output current

C_{IN} input capacitance

C_{OUT} output capacitance

t_R rise time, between 10% and 90% of signal amplitude.

t_F fall time, between 90% and 10% of signal amplitude.

4.2.2 I/O

This contact is used for data exchange as input or output.

Electrical Characteristics

SYMBOL	CONDITIONS		MIN	MAX	UNIT
V_{IH}	Either	I_{IH} max = - 500 uA	2	Vcc	V
	Or(1)	I_{IH} max = ± 20 uA	0.7 * Vcc	Vcc - 0,3	V
V_{IL}		I_{IL} max = - 1mA	- 0.3	0.8	V
V_{OH} (2)	Either	I_{OH} max = - 100 uA	2.4	Vcc	V
	or	I_{OH} max = ± 20 uA	3.8	Vcc	V
V_{OL}		I_{OL} max = 1 mA	0	0,4	V
C_{IN}, C_{OUT}				30	pF
t_R , t_F				1	us

(1) For the interface device, take into account both conditions.

(2) It is assumed that a pull up resistor is used in the interface device (recommended value: 20 K ohms)

When the two ends of the line are in reception mode, the line shall be maintained in the high state.

When the two ends are in non-matched transmit mode, the logic state of the line may be indeterminate. During operation, the interface device and the card shall not both be in transmit mode.

Note: I/O thus has two steady states (as defined in ISO 1177):

- state Z (mark) or high state, if the card and the interface device are in reception mode or if this state is imposed by the transmitter.

- state A (space) or low state, if this state is imposed by the transmitter.

4.2.3 VPP

Idle state : Vcc ± 5 %, 20 mA max.

Active state : P volts ± pa %, I value: See note 3.
pa = programming voltage accuracy, see 6.1.4.3.

Rise or fall time : 200 us max. The rate of change of VPP shall not exceed 2 volts/us.

Note 1: the card provides the interface device with the values of P, pa and I. Default values are P= 25 V, pa = 4 %, I max = 50 mA.

Note 2: the measurements of these figures shall be made in the range of 25 ± 5°C

Note 3: maximum values of I are the following:

00	25 mA
01	50 mA
10	100 mA
11	200 mA

4.2.4 - CLK

The actual frequency, delivered by the interface device on CLK, is designated by f_i or f_s. It may not be the same during the answer to reset (f_i) and during subsequent transmission (f_s).

Electrical Characteristics

SYMBOL	CONDITIONS	MIN	MAX	UNIT
V_{IH}	I_{IH} max = - 200 uA	2.4	Vcc - 0,3	V
	I_{IH} max = ± 20 uA	0,7 * Vcc	Vcc + 0,3	V
	I_{IH} max = ± 10 uA	Vcc - 0,7	Vcc + 0,3	V
V_{IL}	I_{IL} max = ± 200 uA	- 0,3	0,5	V
C_{IN}			30	pF
t_R, t_F			9% of period with a maximum of 0,5 us.	

Duty cycle for asynchronous operation : between 45% and 55 % of the period during stable operation.

4.2.5 RST

Electrical Characteristics

SYMBOL	CONDITIONS	MIN	MAX	UNIT
V_{IH}	I_{IH} max = - 200 uA	4	Vcc + 0,3	V
V_{IH}	I_{IH} max = ± 10 uA	Vcc - 0,7	Vcc + 0,3	V
V_{IL}	I_{IL} max = ± 200 uA	- 0,3	0,6	V

4.2.6 - VCC

5 V ± 5 %.

Icc, the current consumed by the card, shall remain less than 200 mA.

Note: When Vcc is used for programming instead of Vpp, Icc current shall be less than 300 mA.

5 - OPERATING PROCEDURE FOR IC CARDS

This operating procedure is applicable for every integrated circuit card with contacts.

A transaction with the card is conducted through the consecutive operations:

- Activation of the contacts
- Reset of the card
- Answer to Reset
- Command(s)
- Deactivation of the contacts

Notes: 1. Reset of a card can be initiated by the interface device at its discretion at any time.

2. A connector is inactive when all pins remain between 0 V and 0.4 V, referenced to GND pin for currents less than 1 mA.

3. An active state on VPP shall only be provided and maintained when requested by the card.

5.1 Activation of the Contacts

The electrical circuits shall not be activated until the contacts are positioned with respect to the interface device so as to avoid possible damage to any card meeting these standards.

The activation of the contacts by the interface device consists of the consecutive operations:

RST is in low state, VCC shall be powered, I/O in the interface device shall be put in reception mode, VPP shall be raised to idle state, and CLK shall be provided with a suitable and stable clock (see 4.2.4 - CLK).

5.2 Reset of the Card

See figure 1 and figure 2.

By the end of the activation of the contacts (RST in low state, VCC powered and stable, I/O in reception mode in the interface device, VPP stable at idle state, CLK provided with a suitable and stable clock), the card answering asynchronously is ready for reset:

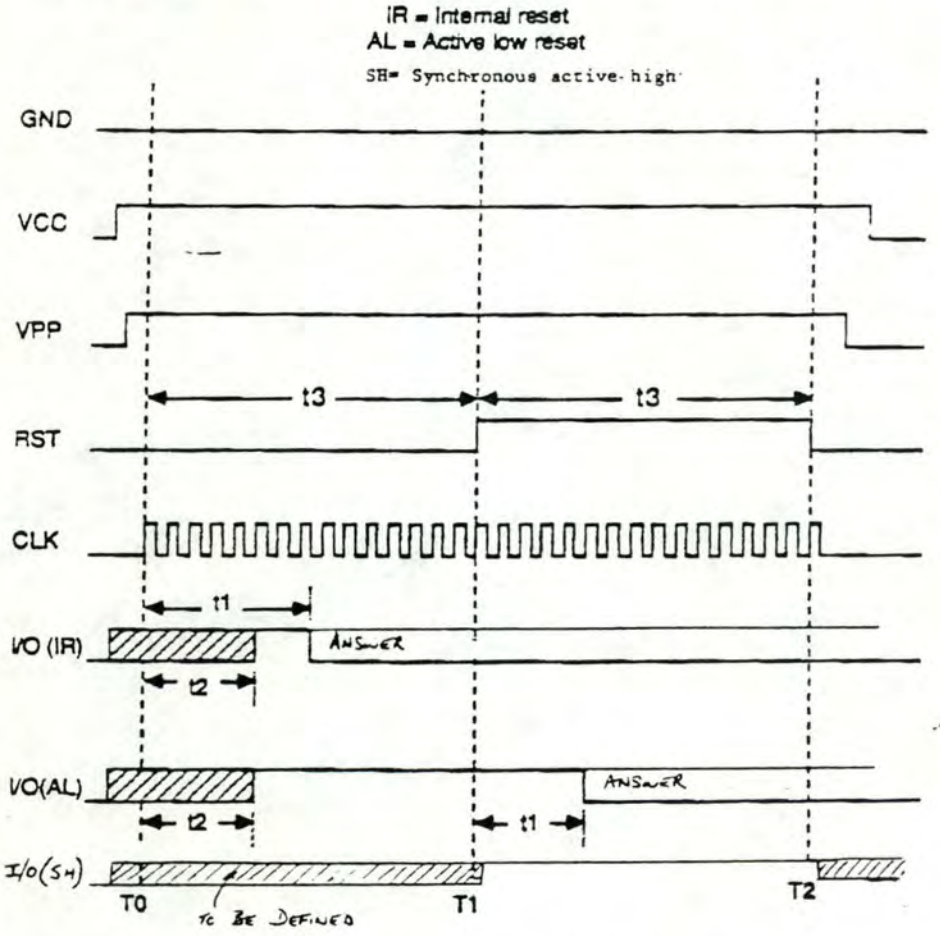
The clock signal is applied to CLK at time T0. The I/O line shall be set to a defined state (high-Z) within 200 clock cycles of the clock signal being applied to CLK (time t2 after T0).

A card with an internal reset is reset after a few cycles of the clock signal. Thus, if the reset is internal, the answer on I/O shall begin between 400 and 40 000 clock cycles after the clock signal is applied to CLK (time t1 after T0).

A card with an active low reset is reset by maintaining RST in low state for at least 40 000 clock cycles after the clock signal is applied to CLK (time t3 after T0). Thus, if no answer occurs within the last 40 000 clock cycles with RST in low state, RST is put to high state (at time T1). The answer on I/O shall begin between 400 and 40 000 clock cycles after the rising edge of the signal on RST (time t1 after T1).

If no answer has been received after 40 000 clock cycles with RST in high state (t3 after T1), the signal on RST shall be returned to low state (at time T2) and the contacts shall be deactivated.

Figure 1 : Examples of reset of the card - Asynchronous Transmission.



$400 \cdot 1/f \leq t1 \leq 40{,}000 \cdot 1/f$

$t2 \leq 200 \cdot 1/f$

$t3 \geq 40{,}000 \cdot 1/f$

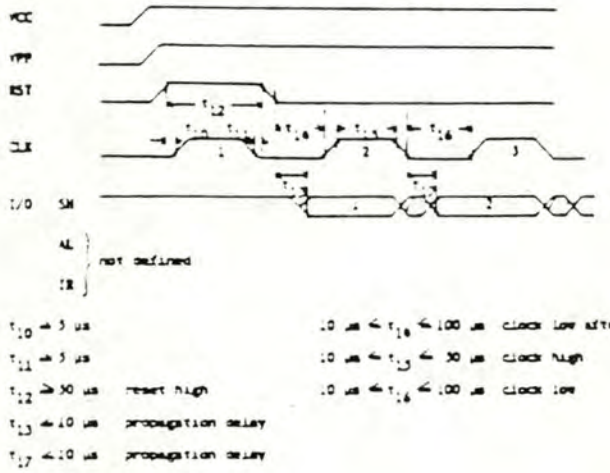
note: the hatched area indicates a period when the state of I/O is undefined .

With a card answering synchronously, the interface device sets all lines to low state. Then VCC is powered, VPP is set to idle state, CLK and RST remain in low state. I/O is put in reception mode in the interface device. A high state shall be maintained on RST for at least 50 microseconds (t_{12}) before returning to low state again (see figure 2).

The clock pulse is applied after an interval (t_{10}) from the rising edge of the reset signal. The duration for the high state of the clock pulse can be any value between 10 us and 50 us and there is only no more than one clock pulse during reset high allowed. The time interval between the falling edges of CLK and RST is (t_{11}).

The first data bit is obtained as an answer on I/O while CLK is low and is valid after an interval (t_{13}) from the falling edge on RST.

Figure 2 Example when a synchronous answer is expected



Note 1 : The internal state of the card is not defined before reset. Therefore the design of the IC card has to avoid improper operation.

Note 2 : In order to continue the transaction with the card, RST shall be maintained in the state where an answer occurs on I/O.

Note 3 : Interface devices may support one or more of these types of reset behaviour. The priority of testing for asynchronous or synchronous cards is not defined in this standard.

5.3 Command(s)

All data exchanged over the I/O circuit correspond to the execution of commands (via RST for reset and via I/O for any other command).

During the processing of a command, the wait for a signal shall not last longer than a maximum delay, named waiting time. Otherwise it shall be assumed that the card or the interface device is unresponsive, showing its disapproval for instance.

As for answer to reset, the operating procedure of commands depend on the type of transmission (asynchronous or synchronous).

5.4 Deactivation of the contacts

When the processing of the last command of the sequence is complete, or when the transaction is aborted (unresponsive card or detection of card removal), the electrical contacts shall be deactivated.

Note: The deactivation by the interface device consists of the consecutive operations: low state on RST, low state on CLK, power off on VPP, state A on I/O, power off on VCC.

6 - ANSWER TO RESET

Two types of transmission are considered:

Asynchronous Transmission

In this type of transmission, characters are transmitted on the I/O line in an asynchronous half duplex mode. Each character codes an 8 bit byte.

Synchronous Transmission

In this type of transmission, a series of bits is transmitted on the I/O line in an half-duplex mode in synchronisation with the clock signal on CLK.

6.1 Answer to Reset in an Asynchronous Transmission

6.1.1 Bit Duration

There is a linear relationship between the Elementary Time Unit (etu) used on I/O and the period provided by the interface device on CLK.

$$\text{initial etu} = \frac{f_o}{f_i} \cdot \frac{1}{9600} \text{ (seconds)}$$ see alternate definition of etu in 6.1.4.1.

f_o = reference frequency (characteristic of a card) in MHz

f_i = initial frequency (provided by the interface device during the answer to reset) in MHz, as defined in 4.2.4.

f_o is the reference frequency required by the card to generate 9600 bits/s.
f_o = 3,579545 MHz.

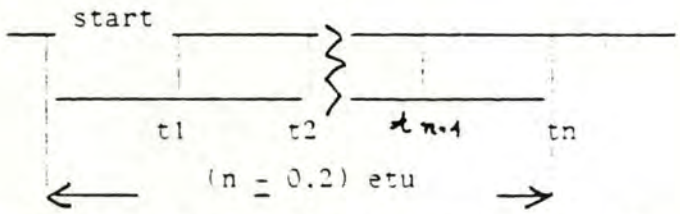
In order to read the initial character (TS), all cards shall initially be operated with f_i in the range of 1 to 4 MHz.

6.1.2 Character Frame

Prior to the transmission of a character, I/O shall be in state Z. A character consists of ten consecutive bits: a start bit in state A, eight bits of information, designated ba to bh and conveying a data byte, and a tenth bit bi used for even parity checking.

Within a character, the time from the leading edge of the start bit to the trailing edge of the nth bit shall equal (n ± 0,2) etu.

The delay between two consecutive characters (between start leading edges) is at least 12 etu, including a character duration (10 ± 0,2) etu plus a guardtime. While in guardtime, the interface device and the card remain both in reception, so that I/O is in state Z.



A data byte consists of 8 bits designated b1 to b8, from the least significant bit (lsb, b1) to the most significant bit (msb, b8). Conventions (level coding, connecting levels Z A to digits (1 or 0) ; and bit significance, connecting ba - bh to b1 - b8) are specified in the first character, called TS, which is transmitted by the card in response to reset. Parity is correct when the number of ONES is even in the sequence from ba to bi.

During the answer to reset, the delay between two consecutive characters from the card shall not exceed 9600 etu, which is equivalent to f_o / f_i seconds.

This maximum value of the delay during answer to reset is named initial waiting time.

6.1.3 Use of Error Detection and Character Repetition

The transmitter tests I/O, (11 ± 0.2) etu after the start leading edge:

- If I/O is in state Z, the correct reception is assumed.
- If I/O is in state A, the transmission was incorrect. The disputed character shall be repeated after a delay of at least 2 etu from error detection.

When searching for a start, the receiver samples I/O periodically. The time origin being the mean between last observation of level Z and first observation of level A, the start shall be confirmed before 0,7 etu, and then ba is received at (1,5 ± 0,2) etu, bb at (2,5 ± 0,2) etu, ... bi at (9,5 ± 0,2) etu. Parity is checked on the fly.

When parity is incorrect, from (10,5 ± 0,2) etu, the receiver transmits an error signal in state A for 1 etu minimum and 2 etu maximum. The receiver then shall expect a repetition of the disputed character.

During answer to reset, this procedure, mandatory for the card, is however optional for the interface device.

Note: When searching for a start, the sampling time shall be less than 0,2 etu so that all the test zones are distinct from the transition zones.

6.1.4 Structure and Content

A reset operation brings about an answer from the card, coded over at most 32 characters : two mandatory characters TS T0, possibly followed by interface characters TAi TBi TCi TDi and "historical" characters T1 T2 ... TK.

The interface characters specify physical parameters of the integrated circuit in the card and logical particularities of the subsequent exchange protocol.

The historical characters designate for example the card manufacturer, the chip inserted in the card, the masked ROM in the chip, the state of the life of the card...

For notational simplicity, T0, TAi ..., T1 ... will designate the bytes as well as the characters in which they are contained.

See annex 1.

6.1.4.1 Structure of TS, the Initial Character

The initial character TS provides a bit sychronization sequence and defines the conventions to code data bytes in the subsequent characters. These conventions refer to ISO 1177.

The two possible values of TS (ten consecutive bits from start to bi, and corresponding hexadecimal value) are:

- A(ZZAAAAAA)Z, '3F' : ONE is A, ba is b8 (msb). Inverse conventions.

- A(ZZAZZZAA)Z, '3B' : ONE is Z, ba is b1 (lsb). Direct conventions.

With I/O initially in state Z, a bit synchronization sequence (Z)AZZA is defined which allows the interface device to determine the etu initially used by the card. An alternate definition of etu is a third of the delay between the first two falling edges in TS. Transmission and reception mechanisms in the card (including the tolerances described in sections 6.1.2 and 6.1.3) shall be consistent with this alternate definition of etu.

6.1.4.2 Structure of the Subsequent Answer to Reset

The subsequent answer to reset contains a variable number of characters in the following order : T0, the format character, TAi, TBi, TCi, TDi, the interface characters, T1,...TK, the historical characters.

The presence of the interface characters is indicated by a bit map technique as explained here after.

The presence of the historical characters is indicated by the number of bytes as specified in the format character.

TO the format character

The TO character contains two parts : the most significant half byte (b8 to b5) is named Y1 and indicates the presence of interface characters, and the least significant half byte (b4 to b1) is named K and indicates the number (0 to 15) of historical characters.

TAi TBi TCi TDi, the interface characters

Bits b5, b6, b7, b8 of the byte containing Yi (TO contains Y1 ; TDi contains Yi - 1) state whether character TAi for b5, whether character TBi for b6, whether character TCi for b7, whether character TDi for b8 are or are not (depending on whether b is 1 or 0) transmitted subsequently in this order after the character containing Yi.

When needed, the interface device shall attribute a default value to an information corresponding to a non transmitted interface character. When TDi is not transmitted, Yi-1 is null.

The interface device shall comply with these requirements in order to process commands.

T1 T2 ... TK, the historical characters

After the last interface character, when K is not null, the answer to reset is continued by K historical characters T1, T2, ... TK.

The specification of the historical characters falls outside the scope of this part of the standard.

The answer to reset is complete at the end of the guardtime of the last character.

6.1.4.3 Specifications of the Interface Characters

Among the interface bytes possibly transmitted by the card in answer to reset, this standard defines only the first 7 : TA1, TB1, TC1, TD1, TA2, TB2, TC2.

These interface bytes convey integer values either equal to or used to compute parameters (as described hereafter) the interface device shall take into account while processing commands.

Integer Values in the Interface Bytes

- TA1 codes FI, sign of M and MI over the most significant half byte (b8 to b5), the bit b4 and the three least significant bits (b3 to b1);
- TB1 codes paI, II and PI1 over the most significant (b8), the next 2 (b7 and b6) and 5 least significant (b5 to b1) bits respectively;

- TC1 codes over the eight bits N, the extra guardtime;
- TD1 codes Y2 and T over the most significant (b8 to b5) and least significant (b4 to b1) half bytes respectively;
- TA2 codes BSI over the eight bits (b8 to b1);
- TB2 codes PI2 over the eight bits (b8 to b1);
- TC2 codes WI, over the eight bits (b8 to b1);

All undefined values of the following parameters are reserved by ISO TC97/SC17 for future use.

Parameters F, D, P, pa, I, N, T, BS, W

F is the clock rate conversion factor for subsequent transmission.
D is the bit rate adjustment factor. The initial etu used during answer to reset is replaced by the work etu during subsequent transmissions in either direction.

During the answer to reset, the bit rate is 9600 bits/s on I/O when f_i provided on CLK is f_o.

During subsequent transmissions, the bit rate is D.9600 bits/s on I/O when f_s provided on CLK is $F.f_o$. The maximum values of f_s are given in table 1 below.

Initial etu : $\frac{f_o}{f_i} \cdot \frac{1}{9\,600}$ (second) Work etu : $\frac{f_o}{f_s} \cdot \frac{1}{9\,600} \cdot \frac{F}{D}$ (second)

f_i, f_s as defined in 4.2.4.

f_o is the reference frequency, characteristic of the card.

P, pa, I define the active state of VPP. Programming voltage : P volts, programming voltage accuracy : pa %, maximum programming current : I mA.

N is an extra guardtime. Before transmitting the next character, the interface device shall ensure a delay of at least (12 - N) etu from the start leading edge of the previous character.

T is a protocol type, summarizing the rules to be used to process commands.

T = 0 is the protocol described in addendum 1 ;
T = 1 is a block transmission protocol, under study as addendum 2 ;
T = 2 and T = 3 are reserved for future full duplex operations.

In a block transmission protocol (T = 1) BS indicates the maximum block size.

W is the waiting time adjustment factor. The initial waiting time used during the answer to reset is replaced by the work waiting time during subsequent commands. Initial waiting time : 9600 etu, also equal to f_o/f_i second. Work waiting time : 9600.D.W etu, equal to $F.(f_o/f_s).W$ seconds.

Default values of these parameters are:

F = D = W = 1; P = 25 volts; pa = 4 % ; I = 50 mA ; N = T = 0 ; BS = 04.

Tables

The correspondance between the parameters F, D, P, pa, I, N, T, BS, W and the integer values FI, DI, PI1, PI2, paI, II, NI, TI, BSI, WI is given in the following tables.

1. F, the clock rate conversion factor

FI	0	1	2	3	4	5	6	7 to 15
F	internal clock	1	1,5	2	3	4	5	RFU
f_s (max) MHz	-	4	6	8	12	16	20	RFU

2. D, the bit rate adjustment factor

$D = 2^M$

MI	0	1	2	3	4	5	6	7
M	RFU	0	1	2	3	4	5	6

Sign (b4) : value 0 means -, value 1 means -.

3. P, the programming voltage, and pa, the programming voltage accuracy

PI1 from 5 to 25 gives the value of P in volts. PI1 = 0 indicates that VPP is not connected in the card which generates an internal programming voltage from VCC. Other values of PI1 are undefined.

When PI2 is present, the indication of PI1 should be ignored. PI2 from 50 to 250 gives the value of P in 0,1 volt. Other values of PI2 are undefined.

paI = 0 indicates pa = 4 % ; paI = 1 indicates pa = 2,5 %.

4. I, the maximum programming current

II	0	1	2	3
I mA	25	50	100	200

5. N, the extra guardtime

N codes directly the extra guardtime, from 0 to 255 in etu.

6. T, the protocol type

T codes the protocol type, with the values 0 to 15.

7. BS, the block size

BSI = 0 indicates 256 bytes ; BS is equal to BSI from 1 to 255.

8. W, the waiting time adjustment factor

W = WI/10 (W in the range from 0.1 to 25.5 seconds).

6.2 Answer to Reset in a Synchronous Transmission

6.2.1 Clock frequency and bit rate

There is a linear relationship between the bit rate on the I/O line and the clock frequency provided by the interface device on CLK.

Any clock frequency between 7 KHz and 50 KHz may be chosen for the reset sequence. A clock frequency of 7 KHz corresponds to 7 Kbit/s, and values of the clock frequency up to 50 KHz cause corresponding bit rates to be transmitted.

6.2.2 Structure of the header of the answer to reset

The reset operation results in an answer from the card containing a header transmitted from the card to the interface.

The header has a fixed length of 32 bits and begins with 2 mandatory fields of 8 bits, H1 and H2.

The chronological order of transmission of the information bits shall correspond to bit identification b1 to b32 with least significant bit transmitted first. The numerical meaning corresponding to each information bit considered in isolation is that of the digit:

0 for a unit corresponding to state A (space)
1 for a unit corresponding to state Z (mark)

6.2.3 Timing of the header

After the reset procedure, see section 5.2 and figure 2, the output information will be controlled by clock pulses. The first clock pulse is applied between 10 us and 100 us (t_{14}) after the falling edge on RST to read the data bits from the card. The clock pulses have a high state whose time can be varied between 10 us and 50 us (t_{15}) and a low state with a time between 10 us and 100 us (t_{10}).

The first data bit is obtained on I/O while clock is low and is valid 10 us (t_{13}) at least after the falling edge of RST. The following data bits are valid 10 us (t_{17}) at least after the falling edge of CLK. Each data bit is valid until the next falling edge of the following clock pulse on CLK. The data bits can therefore be sampled at the rising edge of the following clock pulses.

6.2.4 Data content of the header

The header first allows a quick determination of whether the card and the interface device are compatible. Secondly it enables cards with small memory capacities to use a protocol according to this standard.

If there is no compatibility between the header and the interface device, the contacts shall be deactivated according to section 5.4.

6.2.4.1 Data content of field H1

The first field H1 codes the protocol type. The values of the codes and the corresponding protocol types are:

Hexadecimal value	Protocol type
00 and FF	not to be used
01 to FE	each value is assigned by ISO TC97/SC17 to one protocol type

6.2.4.2 Data content of field H2

The second field codes parameters for the protocol type coded in field H1. The values of H2 are to be assigned by ISO TC97/SC17.

6.2.4.3 Data content of the remaining fields

The specifications of the remaining fields fall outside the scope of this part of the standard. The role of these remaining fields is similar to that of the historical characters mentioned in 6.1.4.

ISO/TC 97/SC 17/WG 4 N 260

ANNEX 1

DIAGRAMS RELATED TO ANSWER TO RESET

(This annex does not form part of the standard)

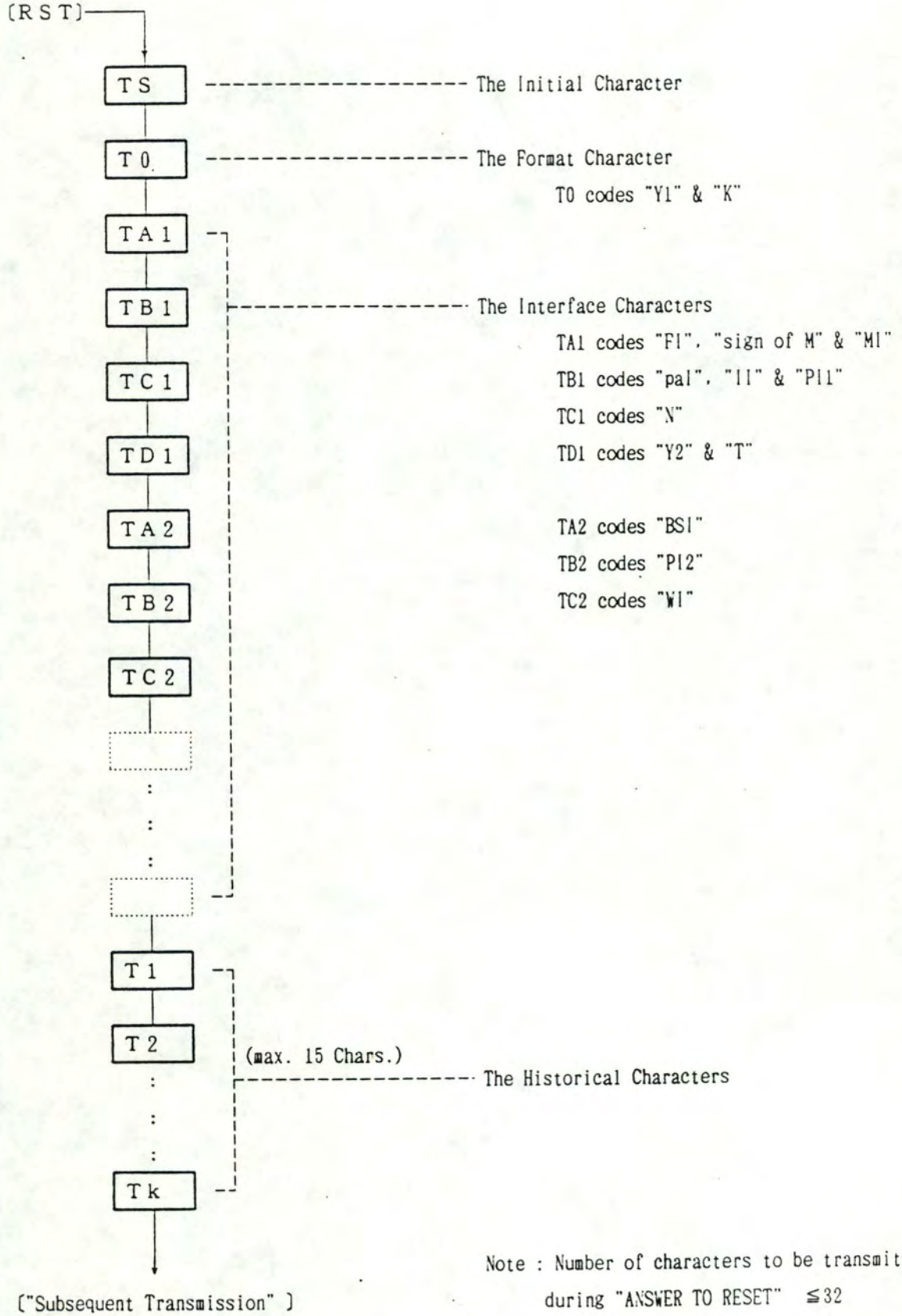
Introductory note :

This annex includes the diagrams on the specification of "ANSWER TO RESET" in an Asynchronous Transmission.

The Protocol Type "T" defined in TDI and all undefined values of the Parameters are reserved by ISO/TC 97/SC 17 for future use.

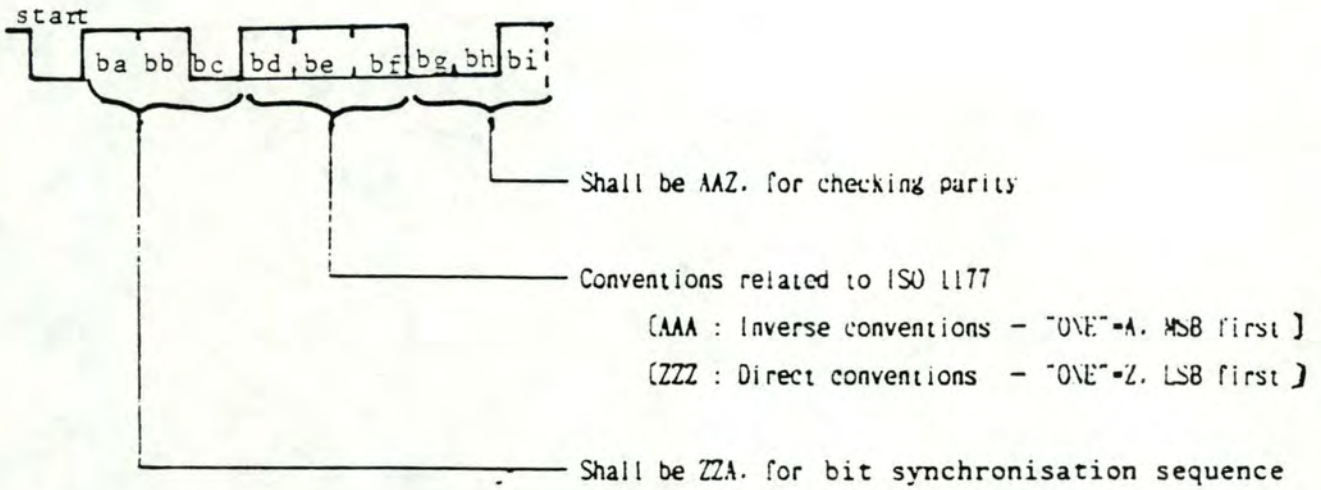
Contents :

I. The Configuration of "ANSWER TO RESET"



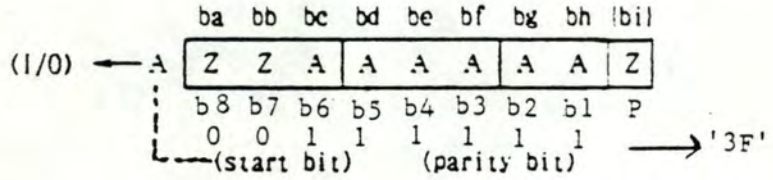
Note : Number of characters to be transmitted during "ANSWER TO RESET" ≦32

II. The Information provided by the Initial Character : TS

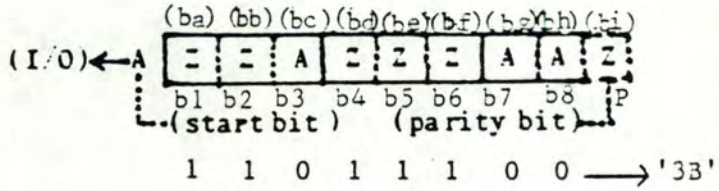


- Following two values are defined. as an internal expression in the card. for reproducing the above information in the interface device.

— Internal value = '3F' : for the cards based on MSB first logic (where "ONE" is A)

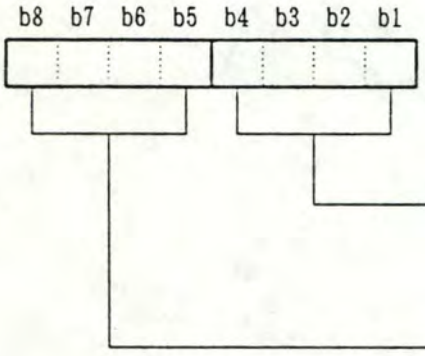


— Internal value = '3B' : for the cards based on LSB first logic (where "ONE" is Z)



Note : The internal expression in the card is assumed for all following diagrams.
The interface device can receive these characters, according to the conventions specified by "TS" (the Initial Character).

Ⅲ. The Information provided by the Format Character : T0 (for "Y1" and "K")



"K" : Number of the Historical Characters
(K = 0~15)

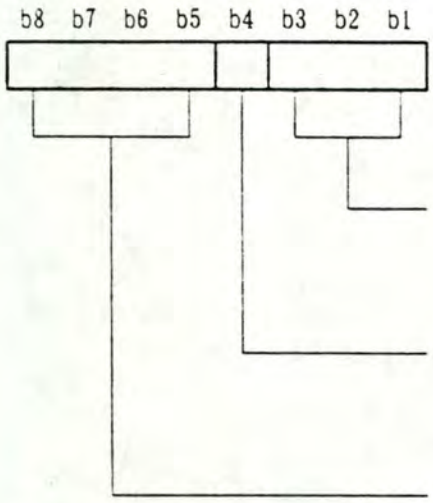
"Y1" : the indicator for the presence of the Interface Characters
(TA1 is transmitted, when b5=1)
(TB1 is transmitted, when b6=1)
(TC1 is transmitted, when b7=1)
(TD1 is transmitted, when b8=1)

Note-1 : The default values are used in the interface device, for the parameters specified in the Interface Characters not transmitted.

Note-2 : b8 shall be 1 for the definition of the Protocol Type "T", execpt when "T" = 0 is selected.

Ⅳ. The Information provided by the Interface Charcters : TAi−TDi

Ⅳ−1 TA1 (The Interface Character for "FI", "sign of M" and "MI")



"MI" : the integer value for the index "M", to obtain the bit rate adjustment factor "D" [default: MI = 1]

"sign of M" : the indicator for the sign of the index "M" [default: sign of M = 0]

"FI" : the integer value for the clock rate conversion factor "F" and for the maximum frequency f_s (max) [default: FI = 1]

[Value of "F" and f_s (max) obtained from "FI"]

"FI"	0	1	2	3	4	5	6	7 to 15
"F"	internal clock	1	1.5	2	3	4	5	RFU
f_s (max) MHz	-	4	6	8	12	16	20	RFU

[Value of "M" obtained from "MI" and "sign of M"] ⟶ $D = (2^M)$

	"sign of M" = 1								"sign of M" = 0						
"MI"	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7
"M"	-6	-5	-4	-3	-2	-1	0	RFU	0	1	2	3	4	5	6

Note-1 : f_s is the frequency provided by the interface device in "Subsequent Transmission", and its maximum value is given by f_s (max) in the above table.

Note-2 : The bit rate adjustment factor "D" is obtained by the formula $D = (2^M)$ and the above table.

Note-3 : Initial etu in second (etu in "ANSWER TO RESET")

$$= (f_0 / f_i) \cdot (1/9600)$$

where f_0 is the reference frequency required by the card to generate 9600 bits/s. and the value is 3.579545 MHz

f_i is the initial frequency provided by the interface device during "ANSWER TO RESET".

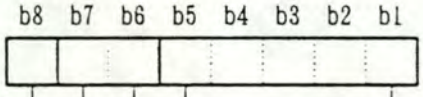
Note-4 : Work etu in second (etu in "Subsequent Transmission")

$$= (f_0 / f_s) \cdot (1/9600) \cdot (F / D)$$

where f_s is the frequency provided by the interface device during "Subsequent Transmission".

Work etu = (Initial etu / D). when $f_s = F \cdot f_i$

Ⅳ-2 TB1 (The Interface Character for "pal", "II" and "PII")



"PII" : the integer value for the active state of the programming voltage, "P" [default: PII = 25]

[P = PII (Volt). where PII = 0 or 5~25]

"II" : the integer value for the maximum current, "I", at the programming voltage [default: II = 1]

[Value of "I" obtained from "II"]

"II"	0	1	2	3
"I" (mA)	25	50	100	200

"pal" : the integer value for the programming voltage accuracy

[pa = ±4%. when pal = 0) [default: pal = 0]

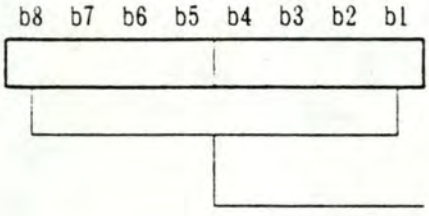
[pa = ±2.5%. when pal = 1]

Note-1 : V_{PP} is generated from V_{CC} internally, when "PII" = 0.

Note-2 : The value of "I" can be obtained by the following formula (for reference).

$I = 25 \cdot (2^{II})$ (mA). where II = 0, 1, 2 or 3

Ⅳ-3 TC1 (The Interface Character for "N")

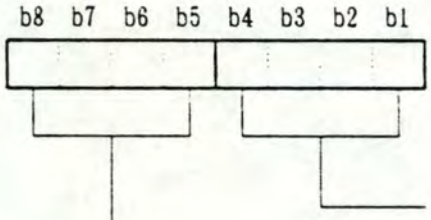


"N" : the extra guardtime assigned by the card to the interface device [default: N = 0]

(minimum delay=12+N (etu) where N = 0~255)

Note : No extra guardtime is assigned by the interface device to the card.

Ⅳ-4 TD1 (The Interface Character for "Y2" and "T")



"T" : the Protocol Type for "Subsequent Transmission" [default: T = 0]

[T = 0 : half-duplex character transmission]

[T = 1 : half-duplex block transmission]

[T = 2 & 3 : full-duplex transmission]

"Y2" : the indicator for the presence of the Interface Characters

[TA2 is transmitted. when b5 = 1]

[TB2 is transmitted. when b6 = 1]

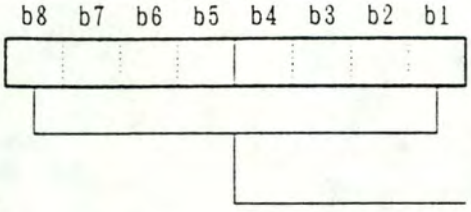
[TC2 is transmitted. when b7 = 1]

[TD2 is transmitted. when b8 = 1]

Note-1 : All other values of "T" are reserved by ISO/TC 97/SC 17 for future use.

Note-2 : "Y2" = 0000 (binary) is assumed. when TD1 is not transmitted.

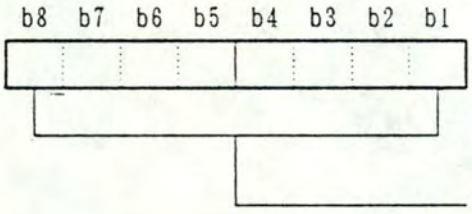
Ⅳ－5 TA2 (The Interface Character for "BSI")



"BSI" : the integer value for the block size, "BS", in bytes
[default: BSI = 64]
[BS = BSI, except when BSI=0]
[BS = 256, when BSI=0]

Note : The value of "BS" indicates the maximum number of characters included in one block.

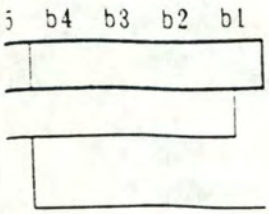
Ⅳ－6 TB2 (The Interface Character for "PI2")



"PI2" : the integer value for the active state of the programming voltage, "P"
[P = 0.1 • PI2 (Volt), where PI2 = 50～250]

Note : The old value of "P", defined by "PI1" in TB1, should be replaced with this value, when "PI2" is present. (The default value is not specified for this parameter.)

(The Interface Character for "Wl")



"Wl" : the integer value for the waiting time adjustment factor "W" . [default: Wl = 10]

[waiting time = F · (f_0 / f_s) · W (second)]

[where W = Wl / 10]

:e delay from the start leading edge of a character to the start leading the following character shall be less than the waiting time:
: at F.fo .

Characters : T1−Tk

specification on theseadditional characters falls outside the scope of this of the standard.

DRAFT PROPOSAL ISO/DP 7816/3 ADD.1	
date 1986 11 10	ISO/TC 97/ SC 17
supersedes document 97/17/4 N 245	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/TC 97 SC 17
" IDENTIFICATION CARDS "

Secretariat B.S.I.

Circulated to P- and O-members, technical committees and international organizations in liaison for :
- discussion at
- comments by
- voting by (P-members only)

Title IDENTIFICATION CARDS - INTEGRATED CIRCUIT(S) CARD WITH CONTACTS
PART 3: ELECTRONIC SIGNALS AND EXCHANGE PROTOCOLS
ADDENDUM 1: STRUCTURE AND PROCESSING OF COMMANDS IN AN ASYNCHRONOUS TRANSMISSION

Introductory note

Addendum 1 - Structure and Processing of Commands in an Asynchronous Transmission

Always initiated by the interface device, a command tells the card what to do in a 5-byte header, and allows a transfer of data bytes under control of procedure bytes sent by the card.

It is assumed that the card and the interface device know a-priori the direction of data, in order to distinguish between instructions for incoming data transfers (where data enter the card during execution) and instructions for outgoing data transfers (where data leave the card during execution). During the process of a command, the procedure of error detection and character repetition described in 6.1.3 is mandatory both in the card and in the interface device.

Interindustry commands for interchange will be defined in a subsequent part of this standard.

1. The command header sent by the interface device

The interface device transmits a header over five successive bytes designated CLA, INS, A1, A2, L.

- CLA is an instruction class.
- INS is an instruction code in the instruction class. The instruction code is valid only if the least significant bit is 0, and the most significant half-byte is neither '6' nor '9'.
- A1, A2 are a reference (eg. an address) completing the instruction code.
- L is the number of data bytes (D1...DL) which are to be transmitted during the command. The direction of movement of these data is a function of the instruction. In an outgoing data transfer command, L=0 introduces a 256-byte data transfer from the card.

After transmission of such a 5-byte header, the interface device waits for a procedure byte.

2. The procedure bytes sent by the card

The values of the procedure bytes request action by the interface device. Three types of procedure bytes are specified:

- ACK (the seven most significant bits in an ACK byte are all equal or all complementary to those in the INS byte, apart from the values '6X' and '9X'). The interface device controls VPP and exchanges data depending on ACK values.
- NUL (='60'). The interface device resets its timer and waits for a new procedure byte, without taking any further action on VPP or on data.
- SW1 (='6X' or '9X', except '60'). The interface device maintains or sets VPP at idle and waits for a SW2 byte to complete the command.

Any transition of VPP (active idle) must occur within guardtime of the procedure byte, or on timer overflow. At each procedure byte, the card can proceed with the command by an ACK or NUL byte, or show its disapproval by becoming unresponsive, or conclude by an end sequence SW1-SW2.

2.1 Acknowledge bytes

The ACK bytes are used to control VPP state and data transfer.

- When exclusive-ORing the ACK byte with the INS byte gives '00' or 'FF', the interface device maintains or sets VPP at idle.
- When exclusive-ORing the ACK byte with the INS byte gives '01' or 'FE', the interface device maintains or sets VPP at active.

- When the seven most significant bits in the ACK byte have the same value as those in the INS byte, all remaining data bytes, DI...DL, if any remain, are transferred subsequently.
- When the seven most significant bits in the ACK byte are complementary to those in the INS byte, only the next data byte, DI, if one remains, is transferred.

After these actions, the interface device waits for a new procedure byte.

2.2 NULL byte (='60')

After resetting its timer, the interface device waits for a new procedure byte.

2.3 Status bytes (SW1 = '6X' or '9X', except '60'; SW2 = any value)

The end sequence SW1-SW2 gives the card status at the end of the command.

The normal ending is indicated by SW1-SW2 = '9000'.

When the most significant half-byte of SW1 is '6', the meaning of the SW1 is independant of the application. Five of these have a special meaning:

'6E' : The card does not support the instruction class
'6D' : The instruction code is not programmed or is invalid
'6B' : The reference is incorrect
'67' : The length is incorrect
'6F' : No precise diagnosis is given

Other values are reserved for future use by TC97/SC17

When the SW1 is neither '6E' nor '6D', the card supports the instruction.

This part of the standard interprets neither '9X' SW1 bytes, nor SW2 bytes: their meaning relates to the application itself.

ANNEX 1

THIS ANNEX DOES NOT FORM PART OF THE STANDARD

Processing of a Command in a Asynchronous Transmission

The command which specifies the function to be executed in the card is issued and transmitted by the interface device. By a procedure sequence the card controls the data transfer. The respond which specifies the result of command execution is issued and transmitted from the card.

1. Structure of Sequence

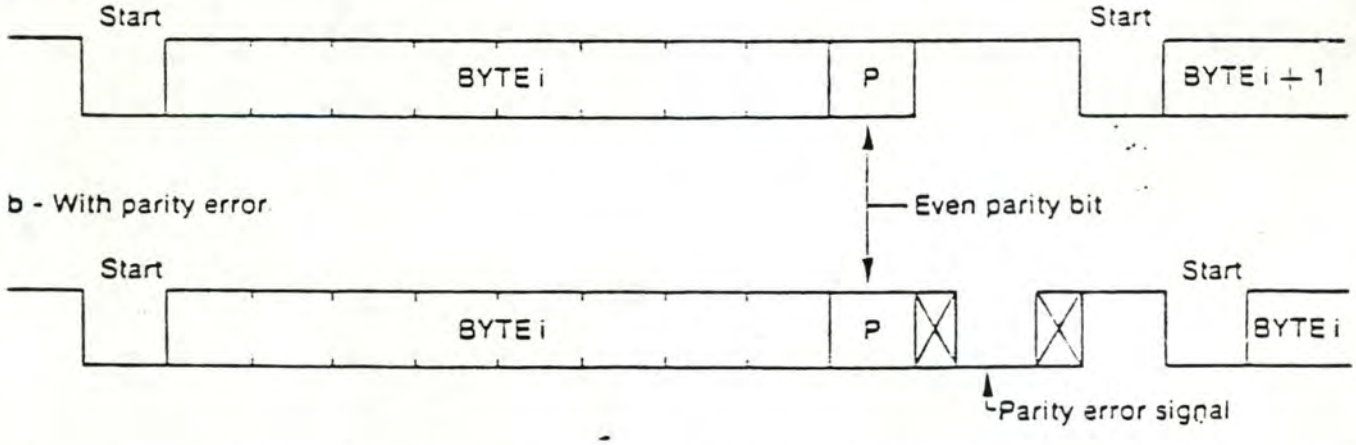
1-1 Basic Structure

A sequence in the asynchronous transmission consists of several fields such as

CLA : Instruction class.
INS : Instruction code.
REF : Reference, consists of two bytes A1, A2.
L : Number of bytes to be transmitted in the Data Field.
PB : Procedure byte.
DATA : Data.
STB : Status byte.

Each field consists of a single byte or several bytes. On the transmission line, a byte in a field is transmitted in the character describe below.

Figure 1 - Byte transmission diagram



Before transmitting the next character, the interface device shall ensure a delay of at least (12 + N) etu from the start leading edge of the previous character.

N is given by the interface byte TC1.

1-2 The Command Block

Interface Device ⟶ Card				
		REF		
CLA	INS	A1	A2	L

1-3 Procedure Sequence

1-3-1 Ack Command Block

C ⟶ ID
PS
ACK

C means Card

ID means Interface Device

1-3-2 Request waiting for a new procedure sequence

C ⟶ ID
PS
NULL

Note : The different values of the procedure byte are specified in section 2.

1-4 Data Sequence

A data transfer is subsequent to an Ack Command Block.

1-4-1 Data Character

ID ⟸ C (1)
DATA
Di

1-4-2 Data Block

Interface Device ⟸ Card (1)			
Data			
Di	Di + 1		DL

(1) The direction of the transfer depends on the Function to be executed.

1-5 End of Command Block

C ⟶ ID	
PB	STB
SW1	SW2

2. Procedure byte and Status code

2-1 Procedure byte

The values of the procedure bytes request action by the interface device

	Value	Action
ACK	INS	Vpp is idle. All remaining data bytes are transferred subsequently.
	INS ⊕ 1	Vpp is activate. All remaining data bytes are transferred subsequently.
	$\overline{INS}$	Vpp is idle. Next data byte is transferred subsequently.
	$\overline{INS \oplus 1}$	Vpp is activate. Next data byte is transferred subsequently.
NULL	60	The interface device waits for a new procedure byte. No further action on Vpp
SW1	SW1	Vpp is idle - the interface device waits for a SW2 byte.

2-2 Status Code

The End of Command Block gives the card status at the end of the command.

When SW1 = 6X, the meaning of SW1 is independant of the application.

Code	Meaning
6E	The card does not support the instruction class
6D	The instruction code is not programmed or is invalid.
6B	The reference is incorrect.
67	The length is incorrect.
6F	No precise diagnosis is given.

The standard interprets neither '9X' SW1 bytes, or SW2 bytes. Their meaning relates to the application itself.

The normal ending is indicated by SW1-SW2 = '9000'

ANNEXE II :
LES CONSIGNES ET LES INSTRUCTIONS

I. Les instructions de contrôle
II. Les instructions de chargement de registres
III. Les instructions de transfert de données
IV. Les instructions arythmétiques et logiques
V. Les utilitaires binaires
VI. Les instructions de branchement
VII. Les instructions d'entrées-sorties cartes
VIII. Les instructions d'entrées-sorties macromachine
IX. Les consignes

I. LES INSTRUCTIONS DE CONTROLE

RESET
RDIR
BOOT
EVENT
ONERR
STOP
ABORT
DELAY
PB
MB
FTYPE
FTAG
DST
BREG
XCHWT
SRC
ENTRY

II. LES INSTRUCTIONS DE CHARGEMENTS DE REGISTRES

LDR
PAD

III. LES INSTRUCTIONS DE TRANSFERTS DE DONNEES

TRR
TR
TRP
TPR
TCP
PST
GST
TPP

IV. LES INSTRUCTIONS ARYTHMETIQUES ET LOGIQUES

ADD
SUB
MUL
DIV
AND
IOR
XOR
NEG
NOT

V. LES UTILITAIRES BINAIRES

SL
SR
RL
RR
CVDB
CVBD
CVAD
CVDA

VI. LES INSTRUCTIONS DE BRANCHEMENT

GOTO
CALL
RET
ACK
REP
IF
BRST
BRPR
EXEC

VII. LES INSTRUCTIONS D'ENTREES-SORTIES CARTE

MST
MHT
INC
OUTC
DICH
SCAN
READ
RDNXT
RDPRV
WRITE
WRNXT
WRPRV

VIII. LES INSTRUCTIONS D'ENTREES-SORTIES MACROMACHINE

IN
OUT
INIT
CRYPT
ES
CNX
DATE

IX. LES CONSIGNES

CM
DLP
DLPK
DLPN
ULP
DLR
KR
ULR
UPD
DEL
CI
RUN
EM

I. LES INSTRUCTIONS DE CONTROLE

-> RESET : cette instruction permet la réinitialisation de la macromachine SCIL .

Rem. L'application paramètre les parties à réinitialiser dans la macromachine :

- soit le contexte de l'interpréteur (pile des appels , ...)
- soit les ressources de la macromachine
- soit la session courante

-> RDIR : cette instruction déclenche l'activation d'un bloc de consignes et provoque si nécessaire l'arrêt de l'interpréteur . Ce bloc est situé à la suite de l'instruction ou est présent à une adresse de la zone programmes .

-> BOOT : cette instruction déclenche l'activation d'une application en session maître après le déroulement d'une ou plusieurs applications intermédiaires .

-> EVENT : cette instruction place le périphérique spécifié sous surveillance . Sur détection d'évènements en provenance du périphérique , un débranchement à une adresse donnée est possible .

-> ONERR : cette instruction permet de gérer un débranchement à une adresse lors d'une erreur interpréteur du type :

- instruction inexistante
- débordement dans la zone programmes
- débordement dans la zone registres
- violation de zone
- erreur de syntaxe
- opération illicite
- erreur périphérique ou carte

-> STOP : cette instruction provoque l'arrêt de l'interpréteur .

-> ABORT : idem , elle peut être activée suite à une erreur rencontrée en l'absence de directive ONERR .

-> DELAY : cette instruction place l'interpréteur en attente d'un signal de tempo fourni par le timer activé (périphérique Di) .

-> P.B. : cette instruction protège un bloc de 2 K se situant dans l'espace mémoire sauvegardée ou résidente de la macromachine (zone programmes) . Seule une routine de niveau 0 peut accéder à un bloc protégé (lecture/écriture) , ce bloc reste exécutable par tous les niveaux .

-> M.B. : cette instruction permet , grâce à son paramètrage , de réaliser toute une liste d'opérations sur les boites aux lettres (Mail Boxes) .

- affectation : allocation dynamique de mémoire à l'application en session, accompagnée d'une présentation de clé .
- désaffectation : désallocation dynamique de mémoire , sur présentation de la clé d'accès .
- ouverture : ouverture d'une boite aux lettres affectée , sur présentation de la clé d'accès .
- fermeture : fermeture d'une boite aux lettres , sur présentation de la clé d'accès .
- réinitialisation : réinitialisation d'une boite aux lettres , sur présentation de la clé d'accès .
- Réinitialisation totale : réinitialisation de toutes les boites aux lettres de la macromachine sur présentation de la clé "maître"

Rem. : il est important de noter que pour chacune des boites aux lettres la macromachine peut gérer jusqu'à huit clés d'accès plus une clé maître .

-> FTYPE : cette instruction recherche l'adresse d'un bloc de type donné (Ex . recherche d'un bloc de surveillance dans le bloc de protocole dans un descripteur de carte) .

-> FTAB : cette instruction recherche , dans la zone programmes , un mot dont le profil est fourni . En cas d'échec de la recherche , un débranchement à une adresse fournie est provoqué .

-> DST : cette instruction permet la mise à jour de la pile de contexte .

Rem. La pile contexte se situe dans la base de travail ou de transfert . Elle est représentée par un bloc de registres dont l'adresse de début est quelconque mais définie .
L'application y charge des paramètres utiles à l'interpréteur pour exécuter des instructions (cfr. RDIR , FTYPE , TRP , TPR , ...) .
L'interpréteur y charge aussi des données résultats suite à l'exécution d'instructions (INIT , ouverture boite aux lettres , ..) .

-> BREG : cette instruction permet à l'application de définir les bases (0-16) qu'elle désire utiliser pour ses bases de transfert et de travail avec le périphérique ou la carte courant .

-> XCHWT : cette instruction donne le rôle de base de travail à la base de transfert et inversément .

-> SRC : lors de cette instruction , le contexte actuel de travail est sauvé dans une pile interne (base de travail et de transfert courantes , chiffrement courant et coupleurs courants) (Cfr. 8.5.7 La base système : 3° partie)

II. LES INSTRUCTIONS DE CHARGEMENTS DE REGISTRES

-> LDR : cette instruction réalise le chargement ou l'initialisation de données dans un bloc de registres de la base de travail courante.

-> PAD : cette instruction initialise , à une même valeur , chaque registre d'un bloc de la base de travail courante .

III. LES INSTRUCTIONS DE TRANSFERT DE DONNEES

Sachant que la macromachine dispose de 16 bases de registres (zone registres) et d'une mémoire adressable (zone programmes) . Il est utile de distinguer les transferts de données internes à la zone registres , internes à la zone programmes et les transferts entre ces deux zones .

1. Internes à la zone registres

-> TRR : cette instruction permet de transférer un bloc de registres de la base de travail courante vers la base de transfert courante .

-> TR : cette instruction réalise le même transfert , mais les paramètres peuvent lui être passés par adresse .

-> TRP : cette instruction permet de transférer un bloc de registres de la base de transfert courante dans la pile contexte (empilage) .

-> TPR : cette instruction permet de transférer un bloc de registres de la pile contexte vers la base de transfert courante (dépilage) .

2. Internes à la zone programmes :

-> TPP : cette instruction permet de transférer des données d'une pile de rangement a une autre .

Rem. L'application est libre de définir et d'utiliser des piles de rangement . Ces piles de rangement se situent dans la zone programmes et sont identifiées par un bloc de mots mémoire dont l'adresse de début est quelconque .

3. Entre la zone registres et la zone programmes :

-> PST : cette instruction permet de transférer un bloc de registres de la base de transfert courante vers une pile de rangement (empilage) .

-> GST : cette instruction permet de transférer des données d'une pile de rangement vers un bloc de registres de la base de transfert courante (dépilage) .

-> TCP : cette instruction permet de transférer les paramètres ISO de l'interface carte courante , présents dans les registres PIC , vers la pile de rangement .

IV. LES INSTRUCTIONS ARYTHMETIQUES ET LOGIQUES

-> ADD : cette instruction réalise l'addition du contenu d'un registre de la base courante de travail avec le contenu d'un registre de la base courante de transfert ou avec une valeur fournie directement .

-> SUB : cette instruction réalise la soustraction de la même manière .

-> MUL : cette instruction réalise la mutiplication de la même manière .

-> DIV : cette instruction réalise la division du contenu d'un registre de la base courante de travail (dividende) avec le contenu d'un registre de la base courante de transfert (diviseur) ou avec une valeur fournie directement .

-> AND : cette instruction réalise un ET Logique entre le contenu d'un registre de la base courante de travail et le contenu d'un registre de la base courante de transfert .

-> IOR : idem , mais OU logique .

-> XOR : idem , mais OU exclusif

-> NOT : cette instruction calcule le complément à 1 du contenu d'un registre de la base courante de travail .

-> NEG : cette instruction calcule le complément à 2 du contenu d'un registre de la base courante de travail .

V. LES UTILITAIRES BINAIRES

-> SL : cette instruction réalise le décalage à gauche (de x positions binaires) du contenu d'un bloc de registres de la base de travail courante . Les bits entrant à droite ont une valeur spécifiée .

-> SR : idem , mais décalage à droite .

-> RL : cette instruction réalise la rotation à gauche (de x positions binaires) du contenu d'un bloc de registres de la base de travail courante . Les bits entrant à droite ont une valeur spécifiée .

-> RR : idem , mais rotation à droite .

-> CVDB : cette instruction réalise la conversion en binaire du contenu d'un bloc de registres de la base de travail courante . Ce contenu initial est supposé contenir des caractères DCB .

-> CVBD : idem , mais contenu initial en binaire et conversion en DCB .

-> CVAD : idem , mais contenu initial en ASCII et conversion en binaire .

-> CVDA : idem , mais contenu initial en DCB et conversion en ACSII.

VI. LES INSTRUCTIONS DE BRANCHEMENT

-> GOTO : cette instruction réalise un débranchement à une adresse donnée .

-> CALL : cette instruction réalise un appel à un sous-programme implanté à une adresse (avec empilage de l'adresse de retour) .

-> RET : cette instruction réalise le retour à partir d'une sous-routine activée par un CALL (avec dépilage de l'adresse de retour) .

-> ACK : cette instruction réalise uniquement le dépilage de l'adresse de retour d'un sous-programme .

-> REP : cette instruction permet l'exécution répétitive d'une suite d'instructions déterminées . Un compteur spécifie le nombre de bouclages à éffectuer .

-> IF : cette instruction réalise un test arithmétique ou logique sur le contenu d'un bloc de la base courante de travail avec le contenu d'un autre bloc de la base courante de travail .

Rem. Le paramètrage de cette instruction permet :

- de fournir une valeur immédiate en lieu et place de ce second contenu ,
- en cas de test réussi , de réaliser un branchement à une adresse ,
- de définir un sens au test :
  réussi si test positif
  réussi si test négatif ,
- de spécifier la nature de la condition directe à évaluer :
  logique vraie
  égalité arithmétique
  inegalité supérieure
  inégalité inférieure ,
- d'effectuer le test du contenu de ce bloc avec une masquage de certains bits .

-> BRST : cette instruction réalise un test sur les mots d'états retournés suite aux instructions-cartes . Ces mots sont : ME1 - ME2 - ME3 présents dans les paramètres d'interface-carte (PIC) de la base de travail courante . Ces trois mots sont testés grâce à une table d'états présente dans la base courante de travail et dont le format est le suivant :

La table comprend une liste de combinaisons à tester et une adresse associée à chaque combinaison .

Les tests sont effectués séquentiellement à partir de la première combinaison . Tant qu'aucune combinaison n'est vérifiée , on teste la suivante et ce jusqu'au bout de la table . Si une combinaison est vérifiée , leur routine présente à l'adresse associée est activée .

-> BRPR : cette instruction généralise le IF de type logique . Elle se base sur le même principe que le test grâce à une table d'état . Le champ testé est quelconque (zone programmes ou registres) , la table utilisée est appelée table de profil et est aussi présente dans la base de travail courante . La table peut donc contenir une liste de descriptions , chaque description contenant le profil du mot à chercher dans le champ et l'adresse du traitement associé .

-> EXEC : cette instruction permet de réaliser un appel à une routine assembleur par un programme SCIL .

VII. LES INSTRUCTIONS D'ENTREES-SORTIES CARTE

-> ST : cette instruction met sous-tension la carte à microprocesseur courante .

-> MHT : cette instruction met hors-tension la carte à microprocesseur courante .

-> INC : cette instruction réalise l'envoi d'un ordre entrant vers la carte à microprocesseur courante .

-> OUTC : idem , mais envoi d'un ordre sortant .

Rem. pour INC et OUTC :

- L'ordre-carte , ses paramètres et ses données (envoyées et reçues) sont contenus dans les registres PIC et DIC des bases de travail et transfert courantes .

- Le profil de l'ordre envoyé peut être surveillé grâce au bloc de surveillance de la carte concernée (cfr descripteur de la carte).

- Les états carte et coupleur ME1 - ME2 - ME3 sont renvoyés dans le registre PIC de la base de travail courante .

-> DICH : cette instruction réalise une recherche d'un mot en progression dichotomique à l'intérieur d'une zone de la carte . Une table de profil nous renseigne sur le mot recherché , et sur les traitements associés .

Rem. : les états carte et coupleur ME1 - ME2 - ME3 sont renvoyés par la carte (PIC) et traités après chaque accès nécessaire à cette recherche dichotomique . Le profil de chaque accès élémentaire peut aussi être surveillé .

-> SCAN : cette instruction réalise une recherche séquentielle de mots à l'intérieur d'une zone de la carte .

Rem. : une table de profils nous renseigne sur le(les) profil(s) du(des) mot(s)-carte recherché(s) et sur leur traitement approprié . Les états carte et coupleur ME1 - ME2 - ME3 sont renvoyés par la carte (PIC) et traités après chaque accès nécessaire à cette recherche séquentielle. Le profil de chaque accès élémentaire à la carte peut aussi être surveillé.

-> READ : cette instruction réalise la lecture d'un mot dans la mémoire de la carte.

-> RDNXT : cette instruction réalise la lecture du mot présent à l'adresse courante de la mémoire carte incrémentée de 1 .

-> RDPRV : cette instruction réalise l'écriture du mot présent à l'adresse courante de la mémoire carte décrémentée de 1 .

-> WRITE : cette instruction réalise l'écriture d'un mot dans la mémoire de la carte .

-> WRNXT : cette instruction réalise l'écriture du mot présent à l'adresse courante de la mémoire carte incrémentée de 1 .

-> WRPRV : cette instruction réalise l'écriture du mot présent à l'adresse courante de la mémoire carte décrémentée de 1 .

Rem. : ces six dernières instructions ne se distinguent guère des ordres de lecture et d'écriture envoyés vers la carte avec l'aide d'ordre INC . Ils n'existent que pour des raisons de comptabilité avec un ancien produit . Chacune des instructions carte est suivie d'un test sur les mots d'états de la carte et du coupleur . La table est donnée dans les paramètres des instructions qui déclenchent un BRST implicite .

VIII. LES INSTRUCTIONS D'ENTREES-SORTIES MACROMACHINE

-> IN : cette instruction permet à la macromachine d'acquérir les données en provenance d'un périphérique .

Rem. la destination des données est soit en zone registre ou en zone programmes . L'acquisition des données est paramètrée par le bloc éditeur dans le descripteur .

-> OUT : cette instruction permet à la macromachine d'émettre des données en direction d'un périphérique et d'enchainer si nécessaire sur une acquisition de données en provenance de ce même périphérique .

Rem. les données émises peuvent être assemblées par le protocole de ligne du driver associé au périphérique émetteur . Inversément , les données reçues peuvent être désassemblées par ce même protocole , présent dans le driver du périphérique récepteur . Les messages prédéfinis dans le bloc éditeur du descripteur du périphérique peuvent être émis .

-> INIT : cette instruction réalise l'initialisation d'un périphérique .

-> CRYPT : cette instruction réalise des opérations de chiffrement , de déchiffrement , de compression et de présentation de clés sur le contenu de registre ou de mémoire .

-> E.S. : cette instruction permet d'activer certains signaux externes de la macromachine (lampes , haut-parleur , ...) .

-> CNX : cette instruction réalise la connexion ou déconnexion d'un périphérique de type terminal ou modem .

-> DATE : cette instruction réalise la lecture ou la mise à jour de la date et l'heure du périphérique horodateur .

IX. LES CONSIGNES

-> CM : cette consigne réalise la mise en mode SCIL de la macromachine .

-> DLP : cette consigne réalise le chargement de programmes . Elle peut être interne ou reçue du périphérique maître .

-> DLPK : cette consigne est identique à DLP , mais réalise en plus le chiffrement

-> DLPN : cette consigne est utilisée suite à DLP , pour charger la suite du programme .

-> ULP : cette consigne réalise l'envoi de données continues dans l'espace programme vers le périphérique maître .

-> DLR : cette consigne réalise le chargement de données dans les registres .

-> KR : idem , mais des données chiffrées , donc activation de l'algorithme de chiffrement courant .

-> ULR : cette consigne réalise l'envoi de données contenues dans les registres vers le périphérique maître .

-> UPD : cette consigne permet l'édition d'un descripteur de périphérique ou de carte .

-> DEL : cette consigne permet l'effacement de données dans un descripteur de périphérique ou de carte .

-> CI : cette consigne réalise l'initialisation d'un périphérique .

-> RUN : cette consigne déclenche l'activation de l'interpréteur . Les instructions exécutées peuvent l'être en mode Debugg .

-> EM : cette consigne émet une fin de mode à la macromachine .